Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

2000-09

Frequency response analysis of T-ACS experimental data.

Elvik, Gary G.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/24293

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NPS ARCHIVE
2000
ELVIK, G.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

FREQUENCY RESPONSE ANALYSIS OF T-ACS EXPERIMENTAL DATA

by

Gary G. Elvik

September 2000

Thesis Advisor: Fotis A. Papoulias

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Frequency Response Analysis of T-ACS Experimental Data	**5. FUNDING NUMBERS**
6. AUTHOR(S) Elvik, Gary G.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

ABSTRACT (maximum 200 words)

Trial runs of a 1:24 scale model crane ship were conducted in the David Taylor Model Basin. The model's response to regular waves under various ship configurations, crane configurations, sea states and ship headings relative to the incoming waves were recorded. The Response Amplitude Operator (RAO) Program analyzes the frequency responses to controlled, regular waves and generates full-scale RAOs as a prediction of the actual ships response. Accurate generation of these full-scale RAOs enables future prediction, using the principle of linear superposition, of ship motions in an irregular sea to be compared to actual, full-scale trial runs being conducted off the coast of California near Camp Pendleton in September 2000.

14. SUBJECT TERMS Power Spectral Density, PSD, Response Amplitude Operator, and RAO	**15. NUMBER OF PAGES** 116
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

FREQUENCY RESPONSE ANALYSIS OF T-ACS EXPERIMENTAL DATA

Gary G. Elvik
Lieutenant, United States Navy
B.S., University of Oklahoma, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2000

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

ABSTRACT

Trial runs of a 1:24 scale model crane ship were conducted in the David Taylor Model Basin. The model's response to regular waves under various ship configurations, crane configurations, sea states and ship headings relative to the incoming waves were recorded. The Response Amplitude Operator (RAO) Program analyzes the frequency responses to controlled, regular waves and generates full-scale RAOs as a prediction of the actual ships response. Accurate generation of these full-scale RAOs enables future prediction, using the principle of linear superposition, of ship motions in an irregular sea to be compared to actual, full-scale trial runs being conducted off the coast of California near Camp Pendleton in September 2000.

TABLE OF CONTENTS

I. INTRODUCTION 1

II. PROBLEM FORMULATION 5

A. THEORY 5
1. Random Process 5
2. Energy Spectra and Power Spectral Densities 5
3. Response Amplitude Operators 6

B. COMPARISON TOOLS 7
1. Spreadsheet Results 7
2. MATLAB Methods 8
a. Welch 8
b. Periodogram 10
c. Multi-taper 11

III. PROGRAM DEVELOPMENT 13

A. GENERAL DESCRIPTION 13
1. Initial Set-up 13
2. Establish Proper String Configuration for the Data Run 15
3. Determine Number of Data Channels to be Processed 15
4. Establish File-path and Load the Raw, Time-based Data 16
5. Compute both Model-Scale and Full-Scale Omega Ranges 17
6. Full-Scale Scalers, PSD Units and RAO Units Library 17
7. Looping/Processing of Data Channels 18
8. Base-lining the Raw, Time-based Data 20
9. FFT Computation 21
10. Model-Scale PSD Computation 21
11. Moving-window Averaging of Model-Scale PSDs 22
12. Computation of Model-Scale RAOs 26
13. Computation of Full-Scale RAOs 27
14. Plot Model-Scale PSDs and Save Figure in jpg Format 27
15. Plot Model-Scale RAOs and Save Figure in jpg Format 28
16. Plot Full-Scale RAOs and Save Figures in jpg Format 29
17. Save Omega Matrices as Last Column of Composite PSD and RAO Matrices 31
18. Save Composite PSD and RAO Matrices 31

B. VALIDATION 32
  1. Load Applicable Matrices 32
  2. Base-line the Raw, Time-Based Data 33
  3. Compute Wave Ht PSDs using MATLAB Methods 33
  4. Compute TACS Roll PSDs using MATLAB Methods 34
  5. Convert MATLAB Method Frequencies to Omegas 34
  6. Compute Areas under the PSD Curves for Comparison 35
  7. Graphical Comparison of Model-Scale PSDs 37
  8. Compute Model-Scale RAOs 41
  9. Graphical Comparison of Model-Scale RAOs 42
  10. Scale to Full-Scale RAOs and Omegas 43
  11. Graphical Comparison of Full-Scale RAOs 44

IV. CONCLUSIONS & RECOMMENDATIONS 47

A. CONCLUSIONS 47

B. RECOMMENDATIONS 48

APPENDIX A. TEST CONFIGURATIONS I 49

APPENDIX B. TEST CONFIGURATIONS II 61

APPENDIX C. TEST CONFIGURATIONS II 63

APPENDIX D. MASTER CHANNEL LIST: CONFIGURATION I 65

APPENDIX E. MASTER CHANNEL LIST: CONFIGURATION II 67

APPENDIX F. MASTER CHANNEL LIST: CONFIGURATION III 69

APPENDIX G. RAO PROGRAM 71

APPENDIX H. PTAVG COMPARISON PROGRAM 89

APPENDIX I. PSD AND RAO COMPARISONS PROGRAM 91

LIST OF REFERENCES 97

BIBLIOGRAPHY 99

INITIAL DISTRIBUTION LIST 101

I. INTRODUCTION

From July 15th to August 17th of 1997, trial runs using 1:24 scale models of a T-ACS Auxiliary Crane Ship, a DDG-963 Class Ship, a Commercial Container Ship and Lighter Barges were conducted in the seakeeping basin of the David Taylor Model Basin at Carderock Division Naval Surface Warfare Center (CDNSWC), Bethesda, Maryland. T-ACS Auxiliary Crane ships are ran by the Military Sealift Command Ready Reserve Force. These self-sustaining, rapidly-deployable ships support military sea transportation needs and are extremely useful in ports that have limited, damaged or undeveloped port facilities. The functions of T-ACS ships are to lift and transfer various loads from either themselves or adjacent vessels and piers.

A total of 251 different model-scale trial runs were performed, each running approximately 8 minutes, under a matrix of variable conditions:

Variable Condition:	Description:	
Ship Configurations	Config I	T-ACS in Center
		Container Ship to Port
		Lighter Barges to Starboard
	Config II	T-ACS in Center
		DDG-963 Class Ship to Port
		Lighter Barges to Starboard

	Config III T-ACS in Center Lighter Barges to Port
Ship Heading	45 degree increments from 0 to 360 degrees relative to incoming waves
Sea State	Model-scale sea states 3, 3 + swell, 4 and 4 + swell
Boom Slew Angle	Angles of 0, 45, 90, 270 and 315 degrees
Rider Block Location	Various positions from 0 to 45.3 degrees
Rider Block Inhaul Angle	Various angles from 0 to 16 degrees
Boom Luff Angle	Angles of 25, 29.6, 54.5 and 60 degrees

A composition of wave height, body motion, velocity and acceleration data for the vessels was recorded for each run from an array of sensors with their locations dependant upon the ship configurations listed above. Results from the 213 Configuration I trial runs, the most probable ship configuration, were provided for frequency response analysis of the raw data to estimate the Full-Scale Response Amplitude Operators (RAOs) of the T-ACS auxiliary crane ship. An RAO is basically a measure or ratio of a vessels response to a regular wave of unit amplitude and thus defined accordingly:

$$RAO = \frac{amplitude\_of\_response}{amplitude\_of\_the\_wave} \qquad \text{(Zubaly,1996,pp. 322)}$$

Tupper expresses the method of studying responses in a seaway in the following manner:

> *"... the apparently random surface of the sea can be represented by the summation of a large number of regular sinusoidal waves, each with its own length, height, direction and phase."*
>
> (Tupper,1996,pp. 104)

> *"... the response of the ship in such a sea could be taken as the summation of its responses to all the individual wave components. Hence the basic building block for the general study of motions in a seaway is the response to a regular simusoidal wave."*
>
> (Tupper,1996,pp. 104)

These concepts strongly identify the need for multiple trial runs using a matrix of variable conditions. The study of responses such as severe rolling in a beam sea or excessive pitching and heaving in a head sea provides extremely valuable insight into a vessels limitations. Operating or load-handling contraints can then be established based upon existing as well as predicted conditions to prevent material damage, downtime and/or, even worse, human injury.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROBLEM FORMULATION

A. THEORY

1. Random Process

Determining RAOs begins with periodic sampling of random processes such as a changing waveheight or a vessels changing responses to the changing waveheight over an interval of time. In this case, each data channel was sampled at a rate of 32.2 Hz over an interval of approximately 8 minutes per data run. This sampling produces the raw data from which the entire RAO determination process is based upon. But, before the raw data can be used, it must be base-lined to remove any offset or bias errors inherent in the sampling system to prevent them from contaminating the process. This is done by determining the mean of the data points for a given channel and then subtracting this mean from each data point. When this step is completed for each data channel, the process is ready to continue.

2. Energy Spectra and Power Spectral Densities (PSDs)

The next phase in determining RAOs is to generate a representative energy spectrum curve for each data channel comprised of Power Spectral Densities (PSDs) vs. wave frequencies (omegas) in which the area under the curve represents the energy associated with the changing data. (Papoulias,2000,pp.109) The PSDs for this curve are derived by expansion of the data points per channel into a Fourier series which approximates the shape of the energy spectrum at the various omegas. (Papoulias,2000,pp. 121) The mean square of the base-lined, raw data record within a narrow band of omega centerd at omega is represented as follows:

$$\bar{r}^2(\omega) = S(\omega)\Delta\omega$$ where

$\bar{r} = data\_mean\_over\_\omega\_band$

$S(\omega) = energy\_spectra$

$\Delta\omega = omega\_band$

(Papoulias,2000,pp. 109)

Thus, as an integrity check along the way which is utilized later in the PSD and RAO comparison porgram, the mean square of the whole, base-lined, raw data record should equal the area under the entire spectral curve:

$$\bar{r}^2(\omega) = \int_0^\infty S(\omega)d\omega$$

(Papoulias,2000,pp. 109)

3. Response Amplitude Operators (RAOs)

Finally, the RAOs are determined by the ratio of the response PSDs over the reference waveheight PSDs as follows:

$$|RAO_R|^2 = \frac{PSD_R}{PSD_W}$$ where

$RAO_R = response\_RAO$

$PSD_R = response\_PSD$

$PSD_W = waveheight\_PSD$

(Papoulias,2000,pp.124)

B. COMPARISON TOOLS

1. Spreadsheet Results

Spreadsheet Results are selected PSDs, RAOs and Omegas for specific Configuration I trial runs that were initially processed by Mr. Dan Hayden of CDNSWC using a spreadsheet program. Only seven data channels were processed per run. These channels were the four pitch and roll channels (channels 3-6) as well as the three relative TACS/Lighter positions (channels 28-30). As for the specific data runs, only 19, 27, 57, 67, 390 and 392 were processed.

The Spreadsheet Results provided a reference to which the RAO Program results would be compared. If the RAO Program results were consistant with each of the seven processed channels, it was assumed that the remaining channels would be correct as well. In order to ensure that the referenced Spreadsheet Results were correct, three additional

MATLAB methods of computing PSDs were used to validate both the Spreadsheet Results and the RAO Program.

2. MATLAB Methods

In addition to the FFT method in the RAO Program, three additional methods of computing PSDs were utilized as comparison tools to ensure validity of both the Spreadsheet Results and the RAO Program: Welch, Periodogram and Multi-taper Methods. The PSDs are computed in units of power per radians while the frequencies in Hz are later converted to radians per second. Although there are many similarities, each method performs the task quite differently.

a. Welch: [PSDs,Freqs] = pwelch(x,nwindow,noverlap,nfft,fs)

The selectable options when utilizing the Welch method are as described as follows:

Option:	Description
x	Base-lined, Raw, Time-based Data Vector
nwindow	Hamming Window Length for Modified Periodograms (Default of [] divides data vector into 8 equal-length windows with residual data points discarded)
noverlap	Number of Overlapping, Windowed Data Points (Default of [] uses 50% overlap from window to window)

nfft	Length of the Fourier Transform for each Window
fs	Data Sampling Frequency (32.2 Hz)

(SPT,welch)

The Welch method produces PSDs by averaging periodograms of overlapping, Hamming-windowed sections of the data vector. Hamming windowing reduces possible sidelobes in the spectral estimate in order to reveal the presence of weaker components of the signal spectrum that may otherwise get hidden. (Marple,1987,pp. 132) Specified-length, discrete Fourier transforms (DFTs) of the overlapping windows are computed as follows:

$$X(k+1) = \sum_{n=0}^{N-1} x(n+1) W_n^{kn} \qquad \text{where} \qquad W_N = e^{-j(2\pi/N)}$$

$$N = length(x)$$

$$X = DFT$$

(SPT,welch)

Indexing of (n+1) and (k+1) is used since MATLAB vectors run from 1 to N instead of 0 to N-1. (SPT,fft) A modified periodogram for each windowed segment is then computed:

$$ModifiedPeriodogram = S(e^{j\omega}) = \frac{\left(\frac{1}{n}\right)\left|\sum_{i=1}^{n} \omega_i x_i e^{j\omega i}\right|^2}{\left(\frac{1}{n}\right)\sum_{i=1}^{n} |\omega_i|^2}$$

(SPT,periodogram)

The individual periodograms are averaged to produce just one representative periodogram. (SPT,periodogram*)* Finally, the PSDs are computed:

$$PSDs = \left(\frac{1}{fs}\right) S_R(e^{j\omega})$$

(SPT,periodogram)

b. Periodogram: [PSDs,Freqs] = periodogram(x,window,nfft,fs)

The selectable options when utilizing the Periodogram method are as follows:

Option:	Description
x	Base-lined, Raw, Time-based Data Vector
window	Window Coefficients for Modified Periodogram of the Input Matrix (Used for "m"x"n" matricies. Default of [] implies single column data vector)

nfft	Length of the Fourier Transform
fs	Data Sampling Frequency (32.2 Hz)

(SPT,periodogram)

The Periodogram method produces PSDs in the same general manner as the Welch method previously discussed. A modified periodogram is generated which leads directly to computation of the PSDs. The differences between the two are that no windowing/segmenting of the data vector occurs and only one periodogram is generated. (SPT,periodogram) Thus, the Welch method is essentially a refinement of the Periodogram method.

c. Multi-taper: [PSDs,Freqs] = pmtm(x,nw,nfft,fs)

The selectable options when utilizing the the third, Multi-taper, method are as follows:

Option:	Description
x	Base-lined, Raw, Time-based Data Vector
nw	Determines Number of Discrete Prolate Spheroidal Sequences $(n = 2_{nw} - 1)$ used as Data Tapers for of Estimation of PSDs (Default of [] is $nw = 4$)

nfft	Length of the Fourier Transform
fs	Data Sampling Frequency (32.2 Hz)

(SPT,pmtm)

The Multi-taper method is by far the most complex of the three comparison methods for production of PSDs. This method combines linear and nonlinear modified periodograms to estimate the PSDs by computing each periodogram using a sequence of orthogonal tapers or windows in the frequency domain as specified from the discrete prolate spheriodal sequences. (SPT,pmtm)

III. PROGRAM DEVELOPMENT

A. GENERAL DESCRIPTION

The RAO program contains a sequence of events utilizing standard MATLAB functions. The validation methods to be discussed later are dependant upon the MATLAB Signals Processing Toolbox from which the commands are comprised of standard functions within the source code. Each significant event of the RAO program is addressed below:

1. Initial Set-up

Before processing a data run, both the raw data disc and the output storage disc must be placed in their respective designated drives. Designation of these drives must conform to the system being used for processing. In order to start the RAO program, the operator enters the desired run number to be processed as a MATLAB command in one of the following two formats:

Command Format:	Applicable DTMB Runs:
rao(##)	19 - 99
rao(###)	101 – 587

Once this has been initiated, the program designates which drive the raw, time-based data is to be read from as well as which drive the generated jpg format figures and composite matrices for the specific run are to be saved to. A floating-point format with 5 digits is placed into effect and the following constants are established:

Constant:	Definition:
ptavg = 50	Number of Sequential PSDs used for Smoothing
window = 500	Number of PSDs within a Rational Freq Range
lambda = 24.175	Actual Model Scale (1:24.175)
freq = 32.2	Raw Data Sampling Frequency in Hz

Selection of the "ptavg" constant is the unavoidable compromise between display resolution and accuracy which will be addressed later. As for the "window" constant, it envelops the first 500 smoothed PSDs which encompass a reasonable range of omegas to be addressed under full-scale conditions. The remaining constant, "freq", is the frequency at which the raw, time-based data was sampled.

```
% Initial  Set-up---------------------------------------
function rao(run)

sread   = char('H:\');
swrite  = char('D:\'

format short e
```

```
ptavg   = 50;
window  = 500;
lambda  = 24.175;
freq    = 32.2;
```

2. Establish Proper String Configuration for the Data Run

Based upon the run number entered by the operator, proper string configurations for the run as well as the folder to which results are to be saved are established. The program then differentiates between a two digit and a three digit run number so that the run string evolves as "R###" and the folder as "###". Broad applicability of designators such as these are utilized throughout the RAO program.

```
% Establish Proper String Configuration for Data Run--
if run <= 99
       s1      = char('R0');
       s2      = char('0');
   else
       s1      = char('R');
end

srun           = num2str(run
filename           = strcat(s1,srun);

if run <= 99
       folder  = strcat(s2,srun);
   else
       folder  = srun;
end
```

3. Determine Number of Data Channels to be Processed

The run number entered enables identification of the number of data channels to be processed due to the particular data channel configuration for the run as follows:

DTMB runs:	Configuration:	Data Channels:
19 – 503	I	46
504 - 533	II	38
534 – 587	III	24

Each data channel is individually labeled per configuration as listed in Appendixes A, B and C.

```
% Determine Number of Data Channels to be Processed---
if run <= 503
      channels  =  46;
   elseif run <= 533
      channels  =  38;
   else
      channels  =  24;
end
```

4. Establish File-path and Load the Raw, Time-based Data

Given the run number and drive designation from which the data is to be read, “filepath” is established as a string in order to execute the MATLAB command “load” to load the raw, time-based data matrix for processing. Once the matrix is loaded, its dimensions are identified utilizing the MATLAB command “size.” The number of columns representing the individual data channels is designated as the variable “channels” and the number of rows representing the number of data points collected is designated as the variable “n”, both of which will be used throughout the program.

```
% Establish Filepath and Load the Raw, Time-based Data
filepath =  [sread,filename];
load(filepath)
szdata   =  size(data);
channels =  szdata(2);
n        =  szdata(1);
```

5. Compute both Model-Scale and Full-Scale Omega Ranges

Before going any further, the Model-Scale omega range is computed based upon the sampling frequency, total number of data points and the "window" length designated earlier. Dividing the Model-Scale range by the square root of the model scale "lambda" produces the Full-Scale omega range. This is supported by the basic assumption for presentation of motion data that "Natural periods of motion vary as the square root of the linear dimension." (Tupper,1996,pp.106) In addition, Tupper identifies that "In watching model experiments the motion always seems rather 'rapid' because of the way period changes . Thus, a 1/25 model will pitch and heave in a period only a fifth of the full-scale ship.

```
% Compute both Model-Scale and Full-Scale Omega Ranges
momega       = 2*pi*freq/n*(0:window-1);
fomega       = momega/sqrt(lambda);
```

6. Full-Scale Scalers, PSD Units and RAO Units Library

There are four individual sets of Full-Scale scalers, PSD units and RAO units in the library associated with the type of data collected by the a particular channel. The combinations are as follows:

Data Type:	Full-Scale Scaler:	PSD Units:	RAO Units:
Wave Height	1	in	dimensionless
Distance	12/lambda	deg	deg/in
Velocity	12/lambda^(3/2)	deg/s	deg/in s
Acceleration	12/lambda^(2)	deg/s^(2)	deg/in s^(2)

```
% Full-Scale Scalers, PSD Units and RAO Units Library-
fs1  = 1;
pu1  = '(in)';
ru1  = '(dimensionless)';

fs2  = 12/lambda;
pu2  = '(deg)';
ru2  = '(deg/in)';
fs3  = 12/lambda^(3/2);
pu3  = '(deg/s)';
ru3  = '(deg/in s)';

fs4  = 12/lambda^2;
pu4  = '(deg/s^2)';
ru4  = '(deg/in s^2)';
```

7. Looping/Processing of Data Channels

It's now time to begin looping to process the "channels" sequentially. From the first data channel to the last, each is assigned a Full-Scale Scaler, "fscaler"; a PSD Units, "psdunits"; an RAO Units, "raounits"; and a Channel Title, "chtitle". These assignments will be utilized later for labeling the generated jpg figures.

```
% Looping/Processing of Data Channels-----------------
for ch = 1:channels

  if channels==46

    if ch==1
        fscaler   =  fs1;
        psdunits  =  pu1;
        raounits  = ru1;
        chtitle   =  'Wave Ht Bow';
      elseif ch==2
        fscaler   =  fs1;
        psdunits  =  pu1;
        raounits  =  ru1;
        chtitle   =  'Sonix Sonic';
      …
      …
      …
      else if ch==46
        fscaler   =  fs4;
        psdunits  =  pu4;
        raounits  =  ru4;
        chtitle   =  'Lghtr1-PBow TvAcc';
      else
    end

  elseif channels==38

    if ch==1
        fscaler   =  fs1;
        psdunits  =  pu1;
        raounits  =  ru1;
        chtitle   =  'Wave Ht Bow';
      elseif ch==2
        fscaler   =  fs1;
        psdunits  =  pu1;
        raounits  =  ru1;
        chtitle   =  'Sonix Sonic';
      …
      …
      …
      elseif ch==38
        fscaler   =  fs4;
        psdunits  =  pu4;
```

```
            raounits  = ru4;
            chtitle   = 'Lghtr1-PBowTvAcc';
         else
      end

   elseif channels==24

      if ch==1
            fscaler   = fs1;
            psdunits  = pu1;
            raounits  = ru1;
            chtitle   = 'Wave Ht Bow';
         elseif ch==2
            fscaler   = fs1;
            psdunits  = pu1;
            raounits  = ru1;
            chtitle   = 'Sonix Sonic';
         ...
         ...
         ...
         elseif ch==24
            fscaler   = fs4;
            psdunits  = pu4;
            raounits  = ru4;
            chtitle   = 'BoomTip-Vert Acc';
         else
      end

   end
```

8. Base-lining the Raw, Time-based Data

The actual processing of data begins here by base-lining or removing the mean from the raw, time-based data. Otherwise, "Failure to remove large sample means or other trends in the data may result in distorted or biased spectral estimates." (Marple,1987,pp.132)

```
% Base-lining the Raw, Time-based Data----------------
x       =  data(:,ch);
x          =  x-mean(x);
```

9. FFT Computation

The MATLAB command "fft" returns the Fourier transform of the base-lined, column data matrix. If the matrix length is a power of two, the "fft" command employs a high-speed fast Fourier transform algorithm. But, this is not always the case since the number of data points per channel varies from run to run In this situation, an alternate mixed-radix algorithm finds the prime factors of the column matrix length then computes the discrete Fourier transforms of the shorter sequences. Either way, the Fourier transform of the base-lined data is produced.

```
% FFT Computation-------------------------------------
Y    =  fft(x,n);
```

10. Model-Scale PSD Computation

The Model-Scale PSD is computed by multiplying the elements of the Fourier Transform by their complex conjugate and dividing by the total number of elements.

```
% Model-Scale PSD Computation-------------------------
PSD  =  Y.*conj(Y)/n;
```

11. Moving-window Averaging of Model-Scale PSDs

This is where the RAO program method differs significantly from the tools in MATLAB previously identified by offering a much finer control over the resolution vs. display accuracy issue. An envelope initially starts with half the "ptavg" window plus one PSD points, averages them together and assigns that average as the first, smoothed PSD point. The envelope then accepts the next sequential PSD point, averages the now larger window and assigns that average as the second, smoothed PSD point. The process continues until the envelope reaches it's maximum size of "ptavg" plus one where the latest average is assigned as the smoothed PSD point for the middle of the envelope. From here on out, the envelope accepts the next sequential PSD point while dropping the first one, thus maintaining a fixed-length, moving window. Refer to the following example:

Example: Let “ptavg” = 4

[..................Freshly Produced PSD Data Points...........................]

[#1 #2 #3 #4 #5 #6 #7 #8 #9 #10]

[1st avg PSD] Window starts at half “ptavg” plus one (i.e. 2 + 1)

[2nd avg PSD] Window increments in size by one

[3rd avg PSD] Max capacity of ptavg + 1 (i.e. 4 + 1)

[4th avg PSD] Window begins shift to the right

[5th avg PSD] Window continues to shift

[6th avg PSD] Window still shifting

[.....................] ...

At this point in the RAO program, the smoothed PSDs are designated as a column of data in a composite PSD matrix with the column designation correlating directly to the data channel being processed. Thus, the first channel processed becomes the first column of PSD data in the composite PSD matrix. As for the chosen window length of 50 plus one, tests of trial runs using incrementally larger point average envelopes yielded a compromise of "ptavg" = 50 which offered the best display resolution while maintaining accuracy in comparison to the spreadsheet results provided (see figs. 1-3)

```
% Moving-window Averaging of Model-Scale PSDs for
  Smoothing
halfptavg = ptavg/2;
sum       = 0;
for k = 1:ptavg
      sum   = sum + PSD(k,1);
      if k > halfptavg
            psd(k-halfptavg,ch) = sum/k;
         else
      end
end
pts          = window + halfptavg;
ptavgplusone = ptavg + 1;
j            = 1;
for k   = ptavgplusone:pts
      sum       = sum + PSD(ptavg+j,1) - PSD(j,1);
      psd(k-halfptavg,ch) = sum/ptavg;
      j         = j + 1;
end

meansqr         = mean(x.^2);
psdarea         = trap(momega,psd(:,ch));
psdscaler       = meansqr/psdarea;
psd(:,ch)       = psd(:,ch)*psdscaler
```

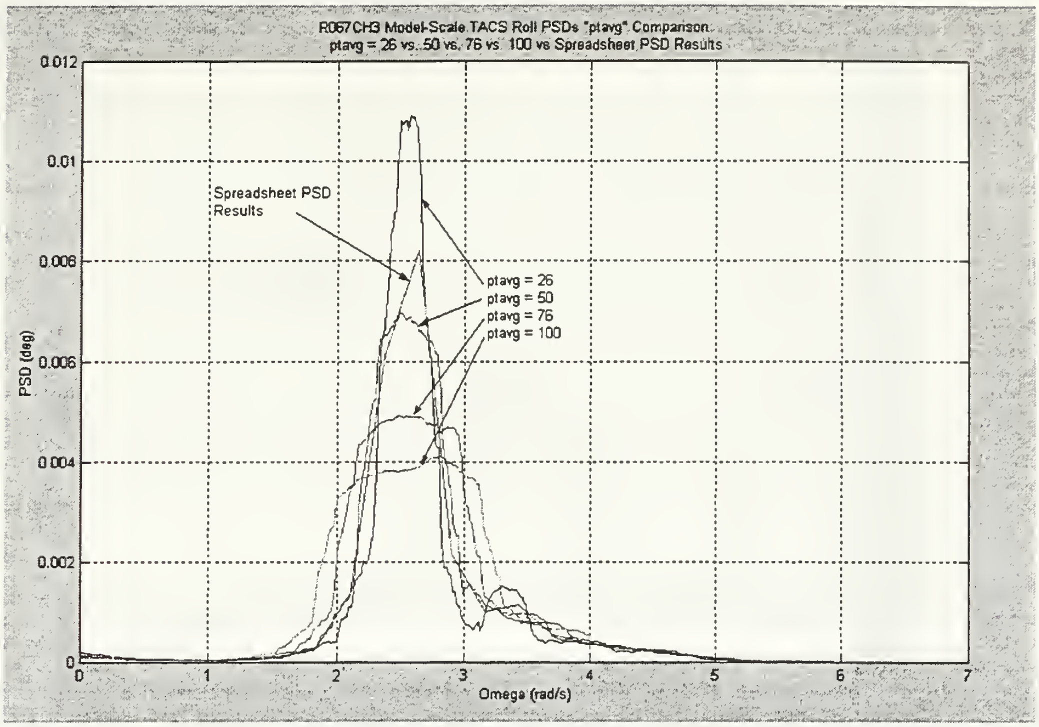


Figure 1.

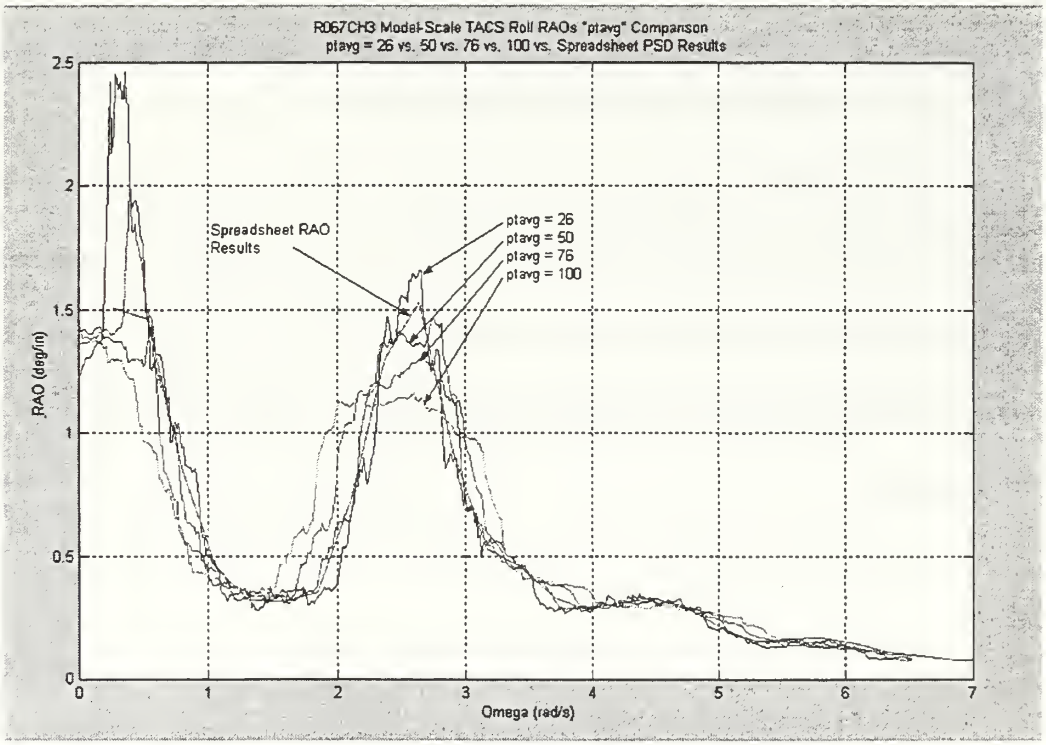


Figure 2.

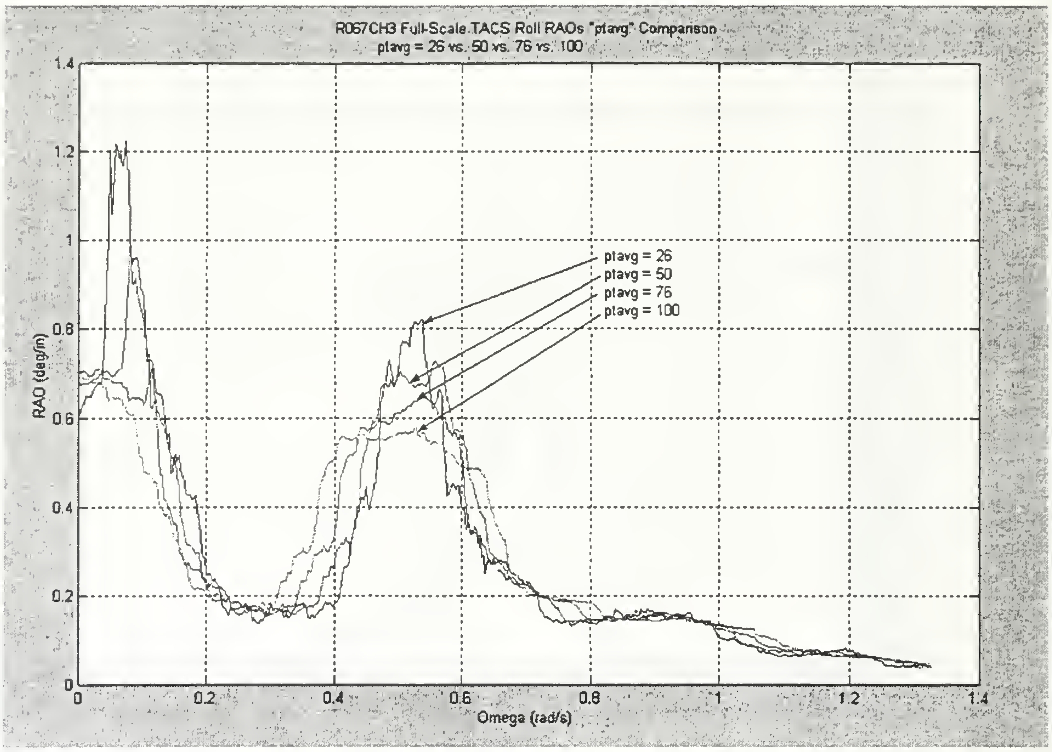


Figure 3.

12. Computation of Model-Scale RAOs

As previously addressed in the theory, RAOs are obtained by taking the individual square roots of the desired PSDs divided by the seaway or wave height PSDs. With the first data channel for each configuration allocated as the reference wave height, the first channel of RAOs processed will naturally be at unity, a constant dimensionless value of one.

```
% Computation of Model-Scale RAOs---------------------
X              =  psd(:,ch)./psd(:,1);
mrao(:,ch)     =  sqrt(X);
```

13. Computation of Full-Scale RAOs

Computation of the Full-Scale RAOs is simply a process of multiplying the Model-Scale RAOs by the Full-Scale Scaler. This previously selected scaler is based upon the nature of the data sampled.

```
% Computation of Full-Scale RAOs----------------------
frao(:,ch)          = mrao(:,ch)*fscaler;
```

14. Plot Model-Scale PSDs and Save Figure in jpg Format

A plot of the smoothed, Model-Scale PSDs vs. their associated Model-Scale omegas is generated (see fig. 4) then saved as a figure in jpg format in a folder dedicated to the particular run and titled accordingly with the run number, channel number and a "p" for Model-Scale PSDs label.

```
% Plot Model-Scale PSDs and Save Figure in jpg Format-
clf
plot(momega(1,:),psd(:,ch))
grid
sch  = num2str(ch);
title(['Model-Scale PSDs for DTMB Run 'srun,' /
   Channel ',sch,': 'chtitle])
xlabel('Omega (rad/s)')
ylabel(['PSD 'psdunits])
saveas(gcf,[swrite,folder,'\',filename,'C',sch,'p'],
   'jpg');
```

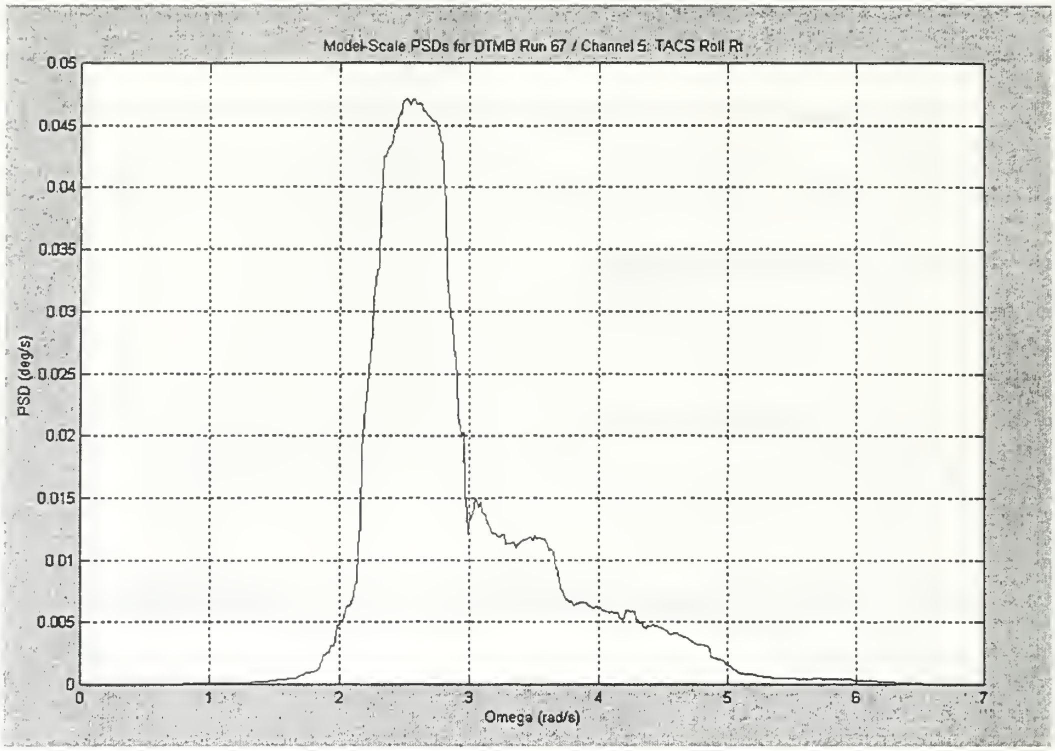


Figure 4.

15. Plot Model-Scale RAOs and Save Figure in jpg Format

A plot of the Model-Scale RAOs vs. their associated Model-Scale omegas is generated (see fig. 5) then saved as a figure in jpg format in a folder dedicated to the particular DTMB run and titled accordingly with the run number, channel number and an “m” for Model-Scale RAOs label.

```
% Plot Model-Scale RAOs and Save Figure in jpg Format-
clf
plot(momega(1,:),mrao(:,ch))
grid
title([’Model-Scale RAOs for DTMB Run ’srun,’ /
   Channel ’,sch,’: ’chtitle])
xlabel(’Omega (rad/s)’)
```

```
ylabel(['RAO 'raounits])
saveas(gcf,[swrite,folder,'\',filename,'C',sch,'m'],
   'jpg');
```

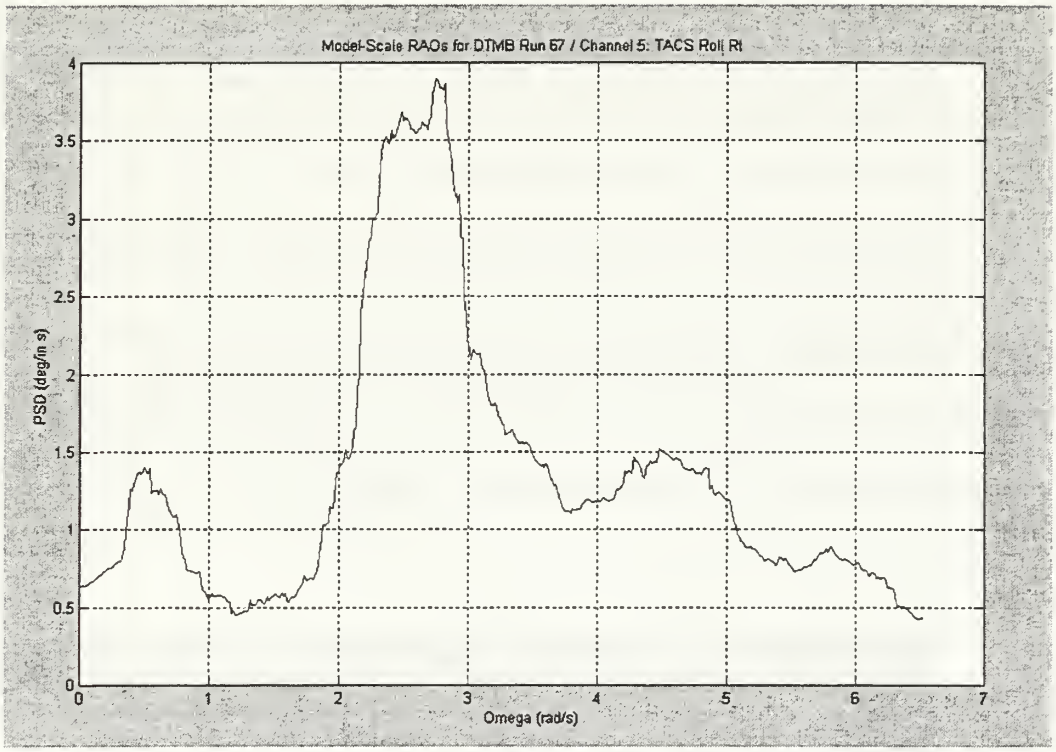


Figure 5.

16. Plot Full-Scale RAOs and Save Figure in jpg Format

A plot of the Full-Scale RAOs vs. their associated Full-Scale omegas is generated (see fig. 6) then saved as a figure in jpg format in a folder dedicated to the particular run and titled accordingly with the run number, channel number and an "f" for Full-Scale RAOs label.

```
% Plot Full-Scale RAOs and Save Figure in jpg Format—
clf
plot(fomega(1,:),frao(:,ch))
grid
title(['Full-Scale RAOs for DTMB Run 'srun,' /
   Channel ',sch,': 'chtitle])
xlabel('Omega (rad/s)')
ylabel(['RAO 'raounits])
saveas(gcf,[swrite,folder,'\',filename,'C',sch,'f'],
   jpg');
```

Processing of the first data channel is now complete. The RAO program then loops back to event 7 listed above, Looping/Processing of Data Channels, until all data channels for the specific run, as determined by the data configuration, have been processed to generate the Model-Scale PSDs, Model-Scale RAOs and Full-Scale RAOs.

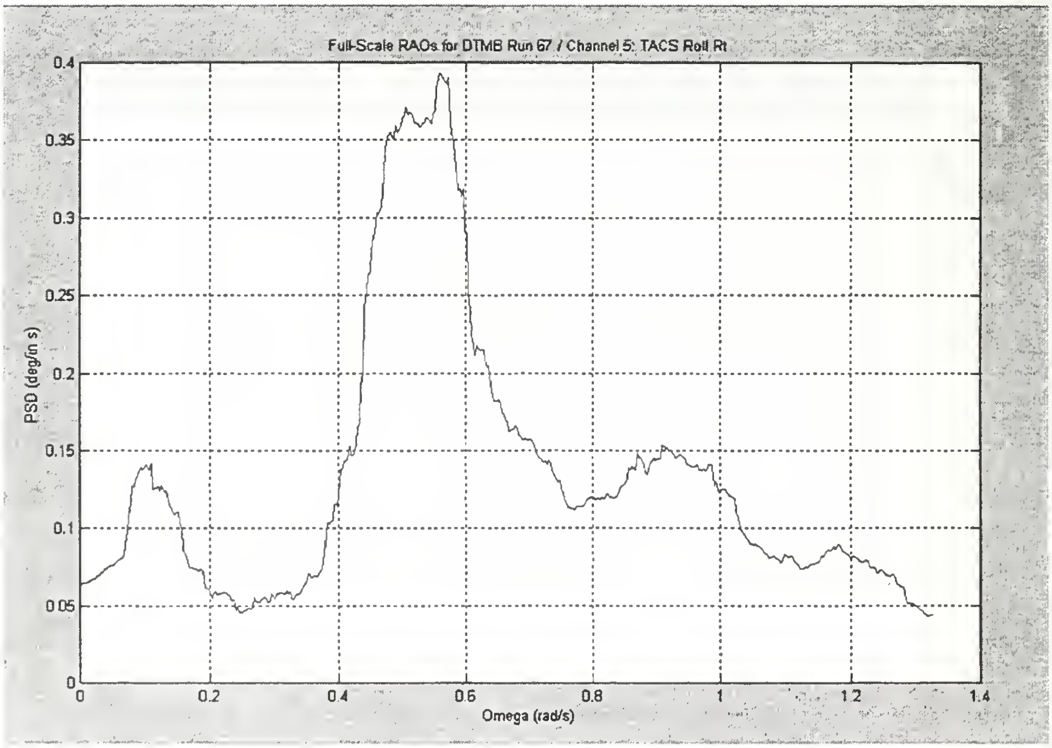


Figure 6.

17. Save Omega Matrices as Last Column of Composite PSD and RAO Matrices

For the purpose of convenient analysis, the single-column omega matrices are tacked on as the final columns of their respective PSD or RAO composite matrices. This basically eliminates the necessity of the jpg formatted figures by allowing direct comparison between any two columns of a composite matrix simply by using a plotting function. Thus, PSD vs. PSD, PSD vs. omega, RAO vs. RAO and RAO vs. omega plots can readily be produced

```
% Save Omega Matrices as Last Column of Composite PSD
  and RAO Matrices
ch           = ch + 1;
psd(:,ch)    = momega(1,:)';
mrao(:,ch)   = momega(1,:)';
frao(:,ch)   = fomega(1,:)';
```

18. Save Composite PSD and RAO Matrices

This final event in the sequence saves the composite PSD and RAO matrices along with the jpg formatted figures previously saved in the folder dedicated to the particular DTMB run. It's now time to process another DTMB run.

```
% Save Composite PSD and RAO Matrices-----------------
save([swrite,folder,'\',filename,'psd.txt'],'psd',
   '-ascii','-tabs');
save([swrite,folder,'\',filename,'mrao.txt'],'mrao',
   '-ascii','-tabs');
save([swrite,folder,'\',filename,'frao.txt'],'frao',
   '-ascii','-tabs');
```

B. VALIDATION

Validation of the RAO Program was performed by utilizing a MATLAB comparison program, “psdandraocomparisons.m”, written to compare the various methods of resultant PSDs; Welch, Periodogram, Multi-taper, Spreadsheet and RAO Program. The comparison program is comprised of a sequence of events with each explained individually as applied to run 67:

1. Load Applicable Matrices

Once the comparison program has been initiated, the following matrices are loaded:

Matrix:	Definition:
R067	Raw, Time-based Data (All 46 Channels)
Run67.txt	Spreadsheet Model-Scale Omegas, PSDs and RAOs
R067psd.txt	RAO Program Model-Scale PSDs and Omegas
R067mrao.txt	RAO Program Model-Scale RAOs and Omegas
R067frao.txt	RAO Program Full-Scale RAOs and Omegas

```
% Load Applicable Matrices-----------------------------
load R067;
load Run67.txt;
load R067psd.txt;
load R067mrao.txt;
load R067frao.txt;
```

2. Base-line the Raw Time-Based Data

Prior to initiating the MATLAB methods of Welch, Periodogram and Multi-taper, the raw, time-based data for both Channel 1: Wave Ht and Channel 3: TACS Roll are base-lined for continuity just as it is done in the RAO Program.

```
% Base-line the Raw Time-based Data from R067---------
x               = data(:,1);     % Wave Ht Data
x               = x - mean(x);

z               = data(:,3);     % TACS Roll Data
z               = z - mean(z);
```

3. Compute Wave Ht PSDs using MATLAB Methods

The Welch, Periodogram and Multi-taper methods are used to compute their Wave Ht PSDs and associated Frequencies. The Frequencies will be converted to Omegas later. For efficiency of computation, a maximum of 8096 FFTs is designated. This happens to be the largest power of two that is less that the total number of data points per channel which range from approximately 14,000 to 16,000. Otherwise, MATLAB uses a slower approach of seeking out the primes of the total number of data points and computing the FFTs according to these segments.

```
% Computation of Wave Ht PSDs and Associated Freqs----
[hwelch,fwelch]   = pwelch(x,[],[],8096,32.2);
[hperio,fperio]   = periodogram(x,[],8096,32.2);
[hmulti,fmulti]   = pmtm(x,[],8096,32.2)
meansqrx          = mean(x.^2);
welchareah        = trap(wwelch,hwelch);
perioareah        = trap(wperio,hperio);
multiareah        = trap(wmulti,hmulti);
hwelchscaled      = hwelch*meansqrx/welchareah;
hperioscaled      = hperio*meansqrx/perioareah;
hmultiscaled      = hmulti*meansqrx/multiareah;
```

4. Compute TACS Roll PSDs using MATLAB Methods

The same process described above is now performed for the TACS Roll PSDs.

```
% Computation of TACS Roll PSDs and Associated Freqs--
[rwelch,fwelch]   = pwelch(z,[],[],8096,32.2);
[rperio,fperio]   = periodogram(z,[],8096,32.2);
[rmulti,fmulti]   = pmtm(z,[],8096,32.2);
meansqrz          = mean(z.^2);
welcharear        = trap(wwelch,rwelch);
perioarear        = trap(wperio,rperio);
multiarear        = trap(wmulti,rmulti);
rwelchscaled      = rwelch*meansqrz/welcharear;
rperioscaled      = rperio*meansqrz/perioarear;
rmultiscaled      = rmulti*meansqrz/multiarear;
```

5. Convert MATLAB Method Frequencies to Omegas

Conversion of frequency to omega is done simply by multiplying the frequency matrices by 2π.

```
% Convert Frequency to Omega---------------------------
wwelch            = 2*pi*fwelch;
wperio            = 2*pi*fperio;
wmulti              = 2*pi*fmulti;
```

6. Compute Areas under the PSD Curves

As an integrity check along the way, the mean squares of the base-lined data are compared to the areas under the PSD curves. At this point, both the Spreadsheet and RAO Program results are integrated into the comparison program to ensure continuity between the various methods with comparative results as follows:

a. Wave Ht Comparison

Method:	Result:
Mean Square of Base-lined Wave Ht Data	0.0124
Welch Area (rad/s)	0.0124
Periodogram Area (rad/s)	0.0124
Multi-taper Area (rad/s)	0.0124
Spreadsheet Area (rad/s)	0.0126
RAO Program Area (rad/s)	0.0124

From the results listed above for run 67, the Welch, Periodogram, Multi-taper and RAO Program methods agree to the nearest 1/1000th when rounded to 5 decimal positions. In addition, each is within 1.6% of the Spreadsheet results.

b. TACS Roll Comparison

Method:	Result:
Mean Square of Base-lined TACS Roll Data	0.0061
Welch Area (deg/s)	0.0061
Periodogram Area (deg/s)	0.0061
Multi-taper Area (deg/s)	0.0061
Spreadsheet Area (deg/s)	0.0059
RAO Program Area (deg/s)	0.0061

From the results listed above for run 67, the Welch, Periodogram, Multi-taper and RAO Program methods agree to the nearest 1/1000th when rounded to 5 decimal positions. In addition, each is within 3.3% of the Spreadsheet results.

```
% Compute Areas under the Wave Ht PSD Curves----------
welchareah2        = trap(wwelch,hwelchscaled);
perioareah2        = trap(wperio,hperioscaled);
multiareah2        = trap(wmulti,hmultiscaled);


spreadareah        = trap(run67(:,2),run67(:,3));
raoprogareah       = trap(R067psd(:,47),R067psd(:,1));

% Compare Areas under the Wave Ht PSD Curves against
     meansqrs
wavehtpsdareacomparison     = [meansqrx welchareah2
     perioareah2 multiareah2 spreadareah raoprogareah]

% Compute Areas under the TACS Roll PSD Curves--------
welcharear2    = trap(wwelch,rwelchscaled);
perioarear2    = trap(wperio,rperioscaled);
multiarear2    = trap(wmulti,rmultiscaled);

spreadarear    = trap(run67(:,2),run67(:,4));
raoprogarear   = trap(R067psd(:,47),R067psd(:,3));

% Compare Areas under the TACS Roll PSD Curves against
     meqnsqrs
rollpsdareacomparison    = [meansqrz welcharear2
     perioarear2 multiarear2 spreadarear raoprogarear]
```

7. Graphical Comparison of Model-Scale PSDs

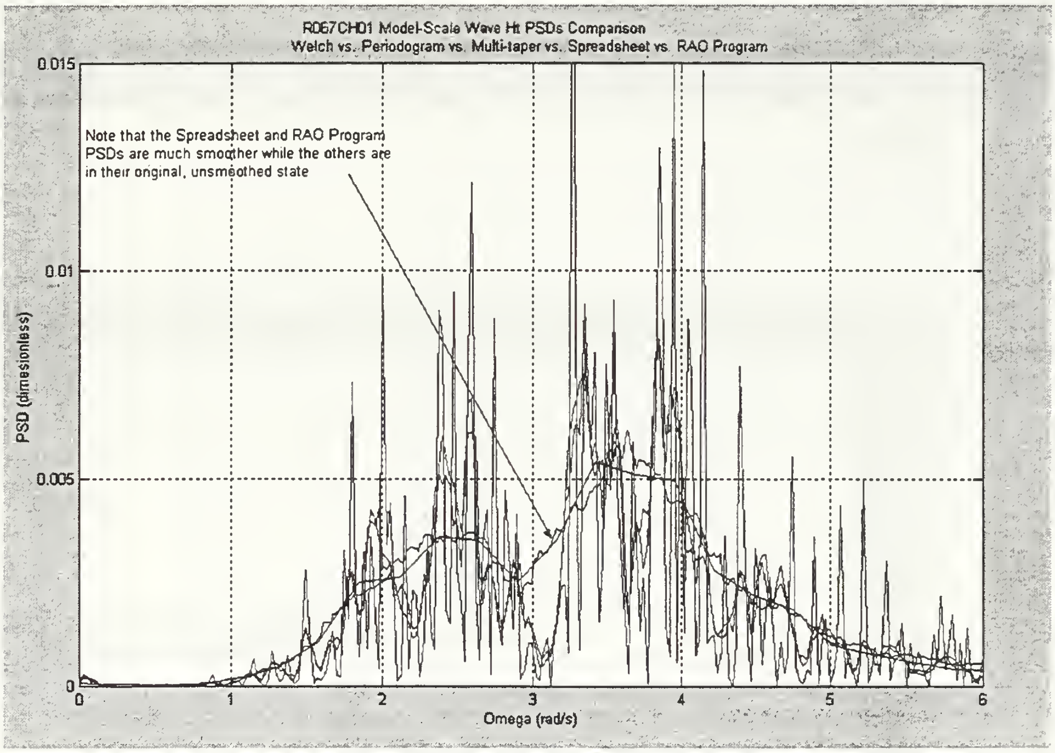
Plots of Model-Scale PSDs from all five methods for both the Wave Ht and TACS Roll are generated for visual comparison. From these comparison plots, it was determined that previously discussed "ptavg" selection of 50 was a very good compromise between display resolution and accuracy when compared to the other four methods of generating PSDs. (see figs. 7-10)

```
% Graphical Model-Scale Wave Ht PSDs Comparison-----
     All 5 Methods
figure(1)
plot(wwelch,hwelchscaled)                   % Welch Method
hold on
plot(wperio,hperioscaled)                   % Perio Method
plot(wmulti,hmultiscaled)                   % Multi Method
plot(run67(:,2),run67(:,3))                 % Spreadsheet
plot(R067psd(:,47),R067psd(:,1));           % Rao Program
title('R067CH01 Model-Scale Wave Ht PSDs Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (dimesionless)')
axis([0 6 0 .015])
grid

% Graphical Model-Scale Wave Ht PSDs Comparison-------
     Spreadsheet vs. RAO Program only
figure(2)
plot(run67(:,2),run67(:,3))                 % Spreadsheet
hold on
plot(R067psd(:,47),R067psd(:,1))            % Rao Program
title('R067CH01 Model-Scale Wave Ht PSDs Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (dimesionless)')
axis([0 6 0 .0065])
grid

% Graphical Model-Scale TACS Roll PSDs Comparison-----
     All 5 Methods
figure(3)
plot(wwelch,rwelchscaled)                   % Welch Method
hold on
plot(wperio,rperioscaled)                   % Perio Method
plot(wmulti,rmultiscaled)                   % Multi Method
plot(run67(:,2),run67(:,4))                 % Spreadsheet
plot(R067psd(:,47),R067psd(:,3));           % Rao Program
title('R067CH03 Model-Scale TACS Roll PSDs
        Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (deg)')
axis([0 6 0 .02])
grid
```

```
% Graphical TACS Roll PSDs Comparison-----------------
     Spreadsheet vs. RAO Program only
figure(4)
plot(run67(:,2),run67(:,4))                % Spreadsheet
hold on
plot(R067psd(:,47),R067psd(:,3))           % Rao Program
title('R067CH03 Model-Scale TACS Roll PSDs
        Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (deg)')
axis([0 6 0 .009])
grid
```



R067CH01 Model-Scale Wave Ht PSDs Comparison
Welch vs. Periodogram vs. Multi-taper vs. Spreadsheet vs. RAO Program

Note that the Spreadsheet and RAO Program PSDs are much smoother while the others are in their original, unsmoothed state

Figure 7.

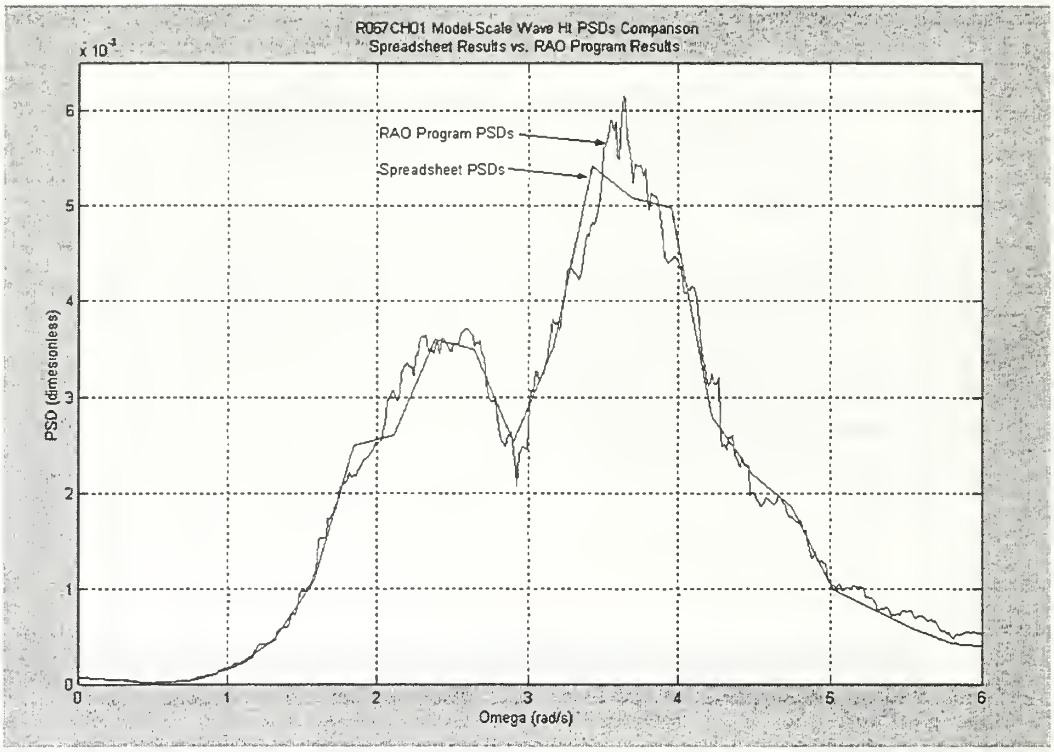


Figure 8.

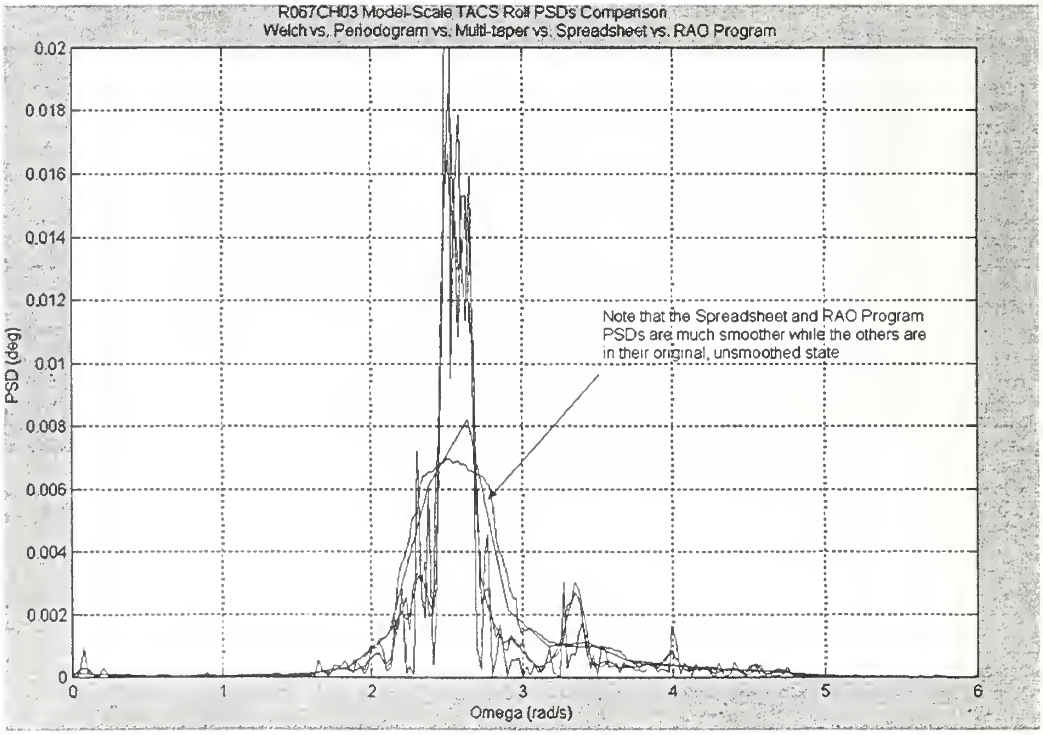


Figure 9.

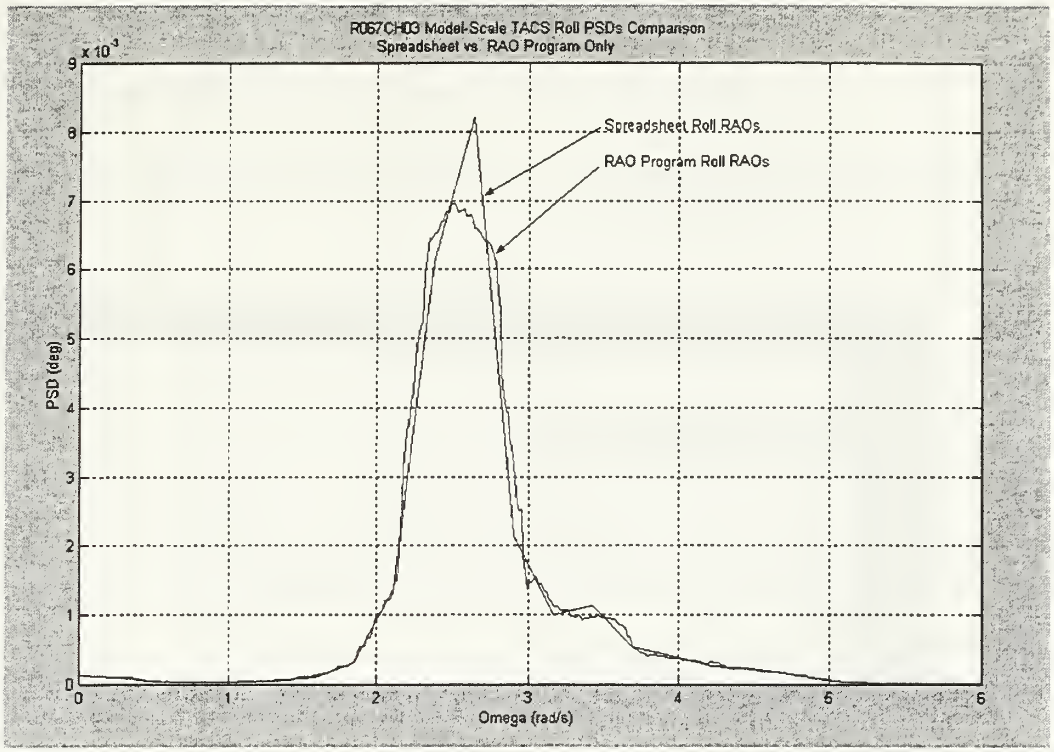


Figure 10.

8. Compute Model-Scale RAOs

Just as it is performed in the RAO Program, Model-Scale RAOs are computed for the Welch, Periodogram and Multi-taper methods by taking the square of the scaled TACS Roll PSDs divided by the scaled Wave Ht PSDs.

```
% Compute Model-Scale TACS Roll RAOs------------------
R1               = rwelchscaled./hwelchscaled;
welchmrao        = sqrt(R1);
R2               = rperioscaled./hperioscaled;
periomrao        = sqrt(R2);
R3               = rmultiscaled./hmultiscaled;
multimrao        = sqrt(R3);
```

9. Graphical Comparison of Model-Scale RAOs

Plots of Model-Scale TACS Roll RAOs from all five methods are generated for visual comparison and they provide very satisfactory results. (see figs. 11-12)

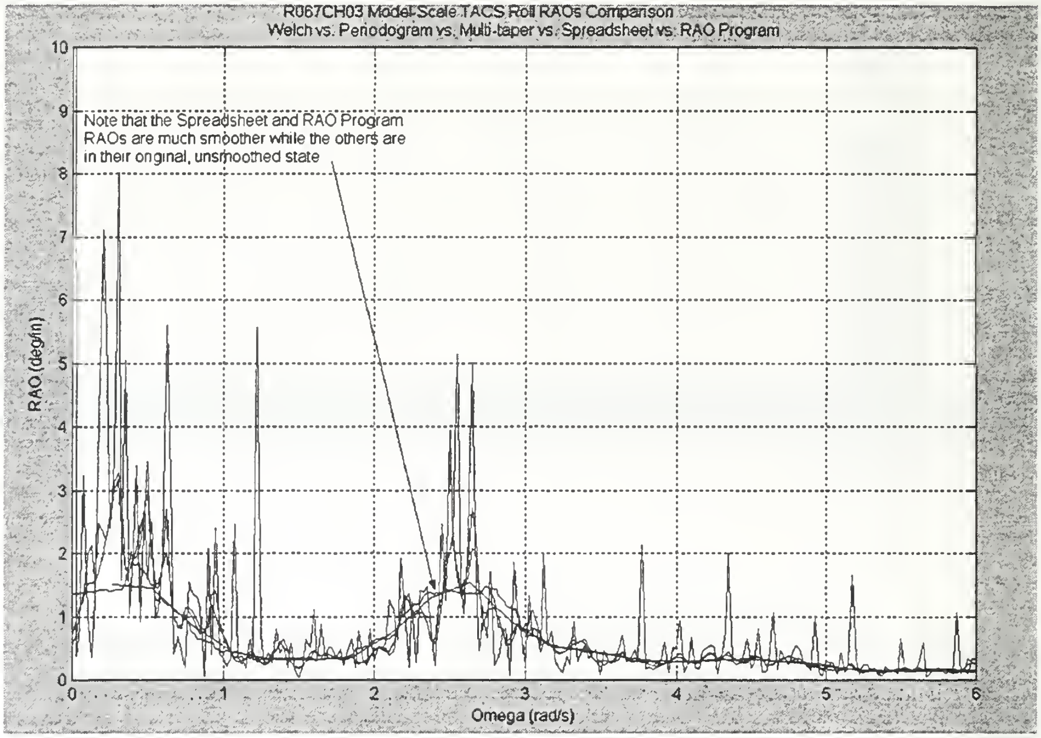


Figure 11.

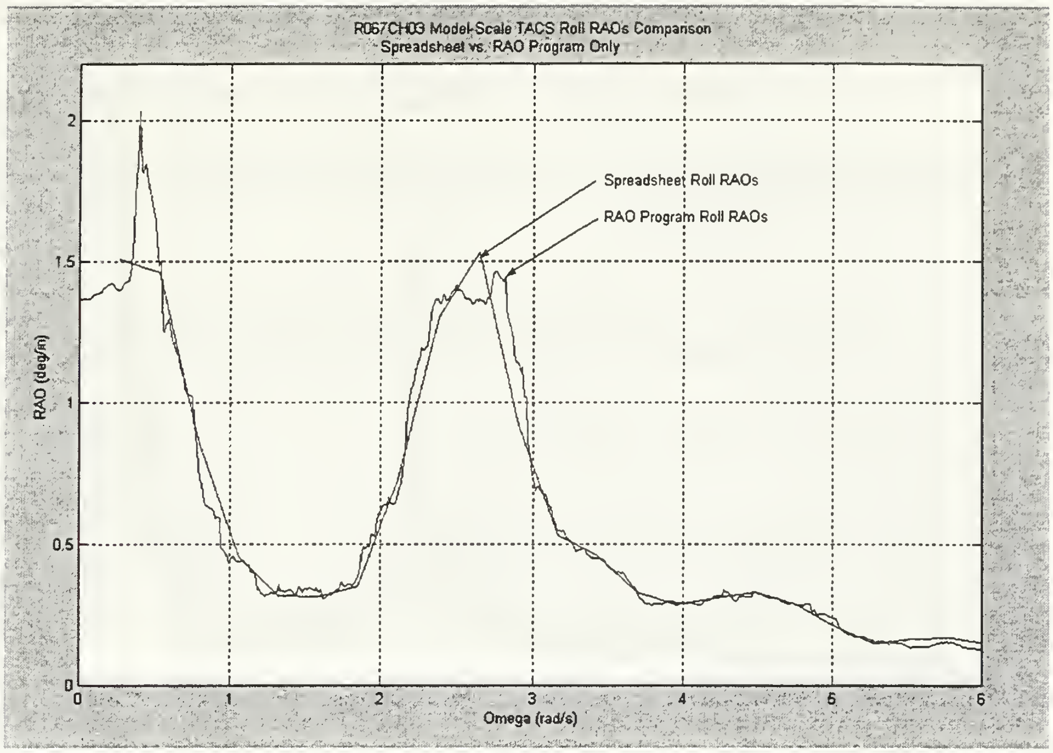


Figure 12.

10. Scale to Full-Scale RAOs and Omegas

In order to compare Full-Scale RAOS and Omegas, the Model-Scale RAOs and Omegas are scaled accordingly just as they are done in the RAO Program.

```
% Scale to Full-Scale RAOs and Omegas-----------------
welchfrao          = welchmrao*12/24.175;
periofrao          = periomrao*12/24.175;
multifrao          = multimrao*12/24.175;

wwelchfull         = wwelch/sqrt(24.175);
wperiofull         = wperio/sqrt(24.175);
wmultifull         = wmulti/sqrt(24.175);
```

11. Graphical Comparison of Full-Scale RAOs

Finally, the Full-Scale RAOs and Omegas for the Welch, Periodogram, Multi-taper and RAO Program methods are plotted for visual comparison. (see figs. 13-14)

```
% Graphical Full-Scale TACS Roll RAOs Comparison------
     4 Methods Not including Spreadsheet
figure(7)
plot(wwelchfull,welchfrao)              % Welch
hold on
plot(wperiofull,periofrao)              % Periodogram
plot(wmultifull,multifrao)              % Multi-taper
plot(R067frao(:,47),R067frao(:,3))      % RAO Program
title('R067CH03 Full-Scale TACS Roll RAOs Comparison')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
axis([0 1.4 0 3])
grid

% Graphical Full-Scale TACS Roll RAOs Presentation----
     RAO Program only
figure(8)
plot(R067frao(:,47),R067frao(:,3))        % Rao Program
title('R067CH03 Full-Scale TACS Roll RAOs
        Presentation')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
axis([0 1.4 0 1.4])
grid
```

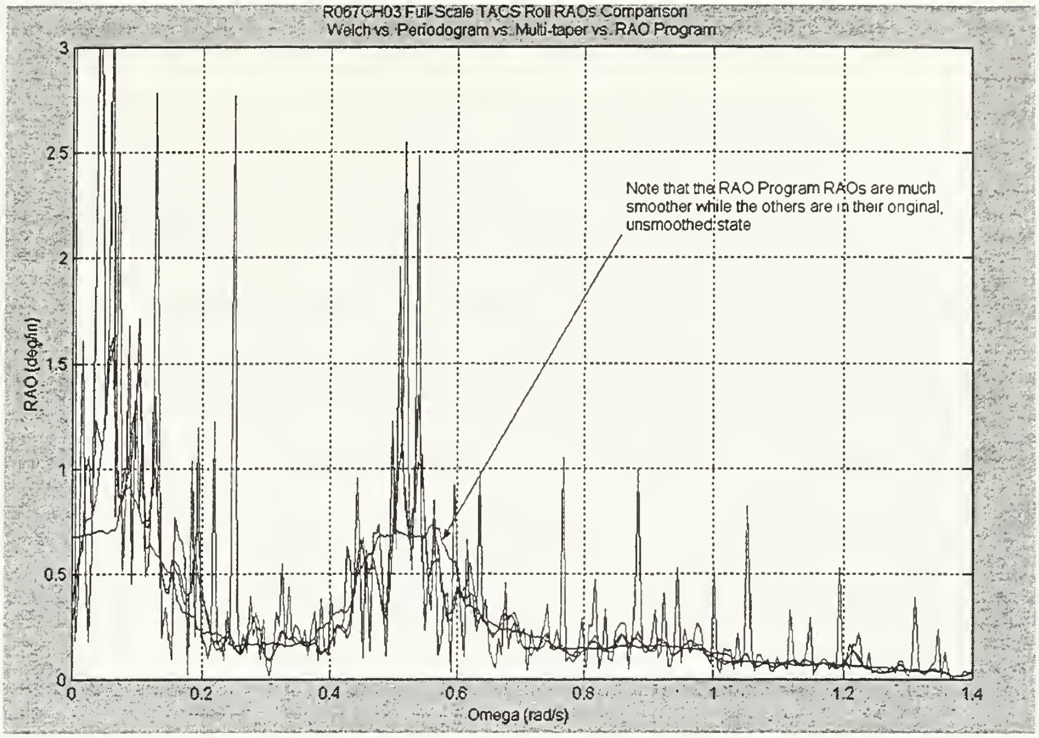


Figure 13.

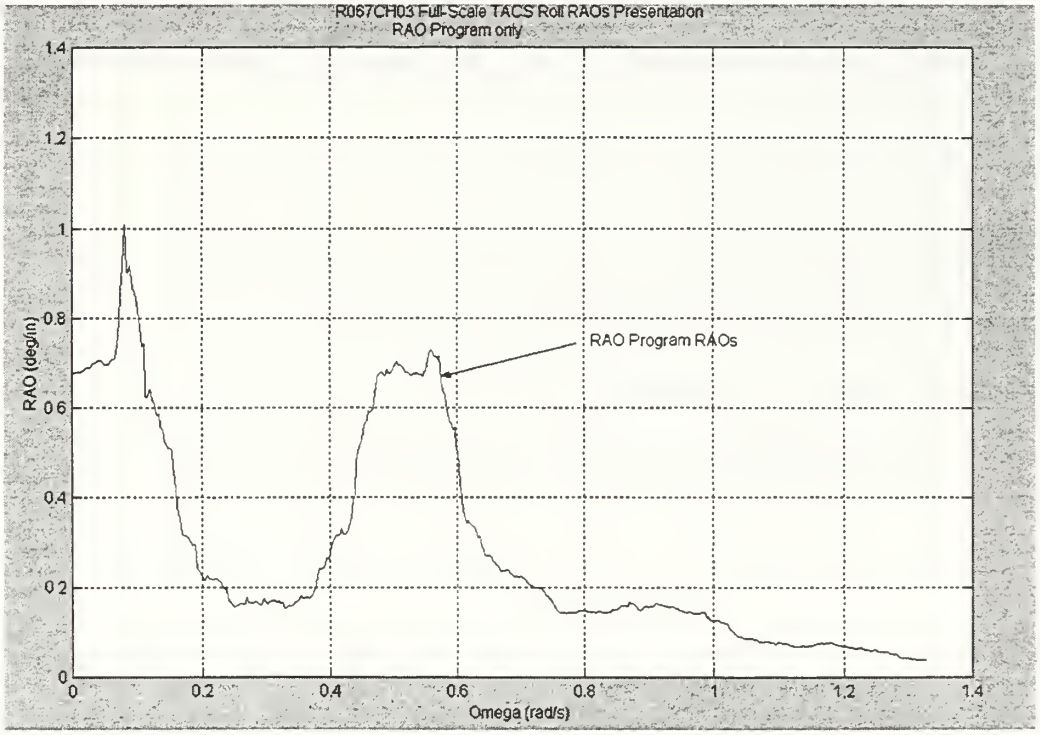


Figure 14.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CONCLUSIONS & RECOMMENDATIONS

A. CONCLUSIONS

Due to the validation comparisons performed on run 67 above, the RAO Program performed quite admirably and accurately. The RAO Programs ability to accurately produce Model-Scale PSDs, Model-Scale RAOs and ultimately Full-Scale RAOs for each channel, along with their associated Omegas, attests to the programs performance. In addition, a short looping program, “raoloop.m”, enabled up to seven data runs to be evaluated sequentially.

```
% Short Program Title: "raoloop.m"--------------------
a  =  [19 21 23 25 27 29 31];     % DTMB Run Numbers
for z = 1:7
   rao(a(1,z));
end
```

The seven run looping limit was imposed due to local hardware configuration limitations. With each data run producing approximately 12.5 megabytes of stored figures and matrices, a 100 megabyte zip disc could safely store up to seven data runs. With a read/write CD drive, the looping capability would have been much greater.

Upon completion of processing the Configuration I runs, the composite matrices (Model-Scale PSDs, Model-Scale RAOs and Full-Scale RAOs) for each run were assembled all on one compact disc as a convenient reference from which any two columns of results may be plotted against each other.

B. RECOMMENDATIONS

An opportunity for a logical continuation of this thesis will soon be available. Trial runs using the actual, Full-Scale ship, the USNS Grand Canyon State (ex-SS President Polk) (T-ACS 3), were conducted September 9th-16th off the coast of Southern California near Camp Pendleton. Data recorded from the Full-Scale trial runs could be processed in a manner similar to the RAO Program and used as a comparison tool to either disprove or further prove the validity of results from the RAO Program. If RAO results from both the at-sea Full-Scale trial runs and the RAO Program model-basin evaluations are in agreement, a multitude of combinations can then be evaluated to predict ship motions in an irregular sea by using the principle of linear superposition.

APPENDIX A: TEST CONFIGURATION I

CDNSWC Run #	Date	Run Storage Directory	Reference Zero Run	NRL Run #	Test Config #	Wave Heading	Sea State	Boom Slew Angle	Rider Block Location	Rider Block Inhaul Angle	Boom Luff Angle	Comments
019	15-Jul-97	TASCA2	017,020	001	i	0	3	0	26.7	0	54.5	
021	15-Jul-97	TASCA2	020,022	002	i	0	3s	0	26.7	0	54.5	
023	15-Jul-97	TASCA2	022,024	003	i	0	4	0	26.7	0	54.5	
025	15-Jul-97	TASCA2	024,026	004	i	0	4s	0	26.7	0	54.5	
027	16-Jul-97	TASCA2	026,028	005	i	90	3	0	26.7	0	54.5	Extreme pendulation perpendicular to boom.
029	16-Jul-97	TASCA2	028,030	006	i	90	3s	0	26.7	0	54.5	Load hit boom & tangled in tag lines.
031	16-Jul-97	TASCA2	030,032	007	i	90	4	0	26.7	0	54.5	Load tangled in tag lines.
033	16-Jul-97	TASCA2	032,034	008	i	0	3	0	0	0	54.5	24 file format
035	16-Jul-97	TASCA2	034,036	009	i	0	3s	0	0	0	54.5	24 file format
037	16-Jul-97	TASCA2	036,038	010	i	0	4	0	0	0	54.5	
039	16-Jul-97	TASCA2	038,040	011	i	0	4s	0	0	0	54.5	
041	16-Jul-97	TASCA2	040,042	012	i	90	3	0	0	0	54.5	
044	16-Jul-97	TASCA2	043,045	014	i	0	3	0	36.5	3.5	54.5	
046	16-Jul-97	TASCA2	045,047	015	i	0	3s	0	36.5	3.5	54.5	
048	16-Jul-97	TASCA2	047,049	016	i	0	4	0	36.5	3.5	54.5	
050	16-Jul-97	TASCA2	049,051	017	i	0	4s	0	36.5	3.5	54.5	
057	16-Jul-97	TASCA2	056,058	018	i	90	3	0	36.5	3.5	54.5	
059	17-Jul-97	TASCA2	058,060	019	i	90	3s	0	36.5	3.5	54.5	

061	17-Jul-97	TASCA2	060,062	020	i	90	4	0	36.5	3.5	54.5	
063	17-Jul-97	TASCA2	062,064	021	i	90	4s	0	36.5	3.5	54.5	
065	17-Jul-97	TASCA2	064,066	022	i	90	2	0	36.5	3.5	54.5	
067	17-Jul-97	TASCA2	066,068	023	i	0	2	0	36.5	3.5	54.5	
069	17-Jul-97	TASCA2	068,070	024	i	0	2	0	26.7	3.5	54.5	
081	17-Jul-97	TASCA2	080,082		i	90	2	0	26.7	3.5	54.5	Repeated as Run 81 due to lost NRL Run 025.
073	17-Jul-97	TASCA2	072,074	026	i	90	2	0	26.7	12.3	54.5	
075	17-Jul-97	TASCA2	074,076	027	i	90	3	0	26.7	12.3	54.5	
077	17-Jul-97	TASCA2	076,078	028	i	90	3s	0	26.7	12.3	54.5	
079	17-Jul-97	TASCA2	078,080	029	i	0	3	0	26.7	12.3	54.5	
081	17-Jul-97	TASCA2	080,082	025	i	90	2	0	26.7	3.5	54.5	24 file format
083	17-Jul-97	TASCA2	082,084	030	i	90	2	0	0	0	54.5	
085	17-Jul-97	TASCA2	084,086	031	i	0	2	0	0	0	54.5	
087	17-Jul-97	TASCA2	086,088	032	i	0	2	270	0	0	54.5	24 file format
089	17-Jul-97	TASCA2	088,090	033	i	90	2	270	0	0	54.5	
091	17-Jul-97	TASCA2	090,092	034	i	90	3	270	0	0	54.5	
093	17-Jul-97	TASCA2	092,094	035	i	90	3	270	0	0	54.5	
095	17-Jul-97	TASCA2	094,096	036	i	90	3	270	0	0	54.5	
098	18-Jul-97	TASCA2	097,099	037	i	0	3	270	0	0	54.5	Lost Barge Rel Mot Dyna Site during run.
100	18-Jul-97	TASCA2	099,101	038	i	90	2	270	26.7	12.2	54.5	
102	18-Jul-97	TASCA2	101,103	039	i	0	2	270	26.7	12.2	54.5	
104	18-	TASCA2	103,105	040	i	90	3	270	26.7	12.2	54.5	

	Jul-97											
106	18-Jul-97	TASCA2	105,107	041	i	0	3	270	26.7	12.2	54.5	
108	18-Jul-97	TASCA2	107,109	042	i	90	3s	270	26.7	12.2	54.5	
110	18-Jul-97	TASCA2	109,111	043	i	0	3s	270	26.7	12.2	54.5	
112	18-Jul-97	TASCA2	111,113	044	i	90	4	270	26.7	12.2	54.5	
114	18-Jul-97	TASCA2	113,115	045	i	0	4	270	26.7	12.2	54.5	
116	18-Jul-97	TASCA3	115,117	046	i	0	2	270	26.7	4	54.5	
118	18-Jul-97	TASCA3	117,119	047	i	90	2	270	26.7	4	54.5	
120	18-Jul-97	TASCA3	119,121	048	i	0	3	270	26.7	4	54.5	
122	18-Jul-97	TASCA3	121,123	049	i	90	3	270	26.7	4	54.5	
124	18-Jul-97	TASCA3	123,125	050	i	0	3s	270	26.7	4	54.5	
126	18-Jul-97	TASCA3	125,127	021	i	90	3s	270	26.7	4	54.5	
128	18-Jul-97	TASCA3	127,129	052	i	0	4	270	26.7	4	54.5	
130	18-Jul-97	TASCA3	129,131	053	i	90	4	270	26.7	4	54.5	
132	18-Jul-97	TASCA3	131,133	054	i	0	4s	270	26.7	4	54.5	
134	18-Jul-97	TASCA3	133,136	055	i	0	2	270	36.5	4	54.5	
135	18-Jul-97	TASCA3	133,136	056	i	90	2	270	36.5	4	54.5	
137	18-Jul-97	TASCA3	136,138	057	i	0	3	270	36.5	4	54.5	
139	18-Jul-97	TASCA3	138,140	058	i	90	3	270	36.5	4	54.5	
141	18-Jul-97	TASCA3	140,143	059	i	0	3s	270	36.5	4	54.5	
142	18-Jul-97	TASCA3	140,143	060	i	90	3s	270	36.5	4	54.5	
144	18-Jul-	TASCA3	143,145	061	i	0	4	270	36.5	4	54.5	

	97											
146	18-Jul-97	TASCA3	145	062	i	90	4s	270	36.5	4	54.5	
150	21-Jul-97	TASCA3	149,151	063	i	0	2	270	36.5	12	54.5	
152	21-Jul-97	TASCA3	151,153	064	i	90	2	270	36.5	12	54.5	
154	21-Jul-97	TASCA3	153,155	065	i	0	3	270	36.5	12	54.5	
156	21-Jul-97	TASCA3	155,157	066	i	90	3	270	36.5	12	54.5	
158	21-Jul-97	TASCA3	157,159	067	i	0	3s	270	36.5	12	54.5	
160	21-Jul-97	TASCA3	159,161	068	i	90	3s	270	36.5	12	54.5	
162	21-Jul-97	TASCA3	161,163	069	i	0	4	270	36.5	12	54.5	
164	21-Jul-97	TASCA3	163,165	070	i	90	4	270	36.5	12	54.5	
166	21-Jul-97	TASCA3	165,167	071	i	0	4s	270	36.5	12	54.5	
168	21-Jul-97	TASCA3	167,169	072	i	90	4	270	36.5	12	54.5	
171	21-Jul-97	TASCA3	170,172	073	i	0	2	270	36.5	12	29.6	
173	21-Jul-97	TASCA3	172,174	074	i	90	2	270	36.5	12	29.6	
175	21-Jul-97	TASCA3	174,176	075	i	0	3	270	36.5	12	29.6	
177	21-Jul-97	TASCA3	176,178	076	i	90	3	270	36.5	12	29.6	
179	21-Jul-97	TASCA3	178,180	077	i	0	3s	270	36.5	12	29.6	
181	21-Jul-97	TASCA3	180,182	078	i	90	3s	270	36.5	12	29.6	
183	21-Jul-97	TASCA3	182,184	079	i	0	4	270	36.5	12	29.6	
185	21-Jul-97	TASCA3	184,186	080	i	90	4	270	36.5	12	29.6	
187	21-Jul-97	TASCA3	186,188	081	i	0	4s	270	36.5	12	29.6	
189	21-Jul-97	TASCA3	188,190	082	i	90	4s	270	36.5	12	29.6	

192	21-Jul-97	TASCA3	191,193	083	i	0	2	0	36.5	11.9	54.5	
194	21-Jul-97	TASCA3	193,195	084	i	90	2	0	36.5	11.9	54.5	24 file format
196	21-Jul-97	TASCA3	195,197	085	i	0	3	0	36.5	11.9	54.5	
198	21-Jul-97	TASCA3	197,199	086	i	90	3	0	36.5	11.9	54.5	
201	22-Jul-97	TASCA3	200,202	087	i	0	3s	0	36.5	11.9	54.5	
203	22-Jul-97	TASCA3	202,204	088	i	90	3s	0	36.5	11.9	54.5	
205	22-Jul-97	TASCA3	204,206	089	i	0	4	0	36.5	11.9	54.5	
207	22-Jul-97	TASCA3	206,208	090	i	90	4	0	36.5	11.9	54.5	DTMB Data lost.
209	22-Jul-97	TASCA3	208,210	091	i	0	4s	0	36.5	11.9	54.5	24 File Format.
211	22-Jul-97	TASCA3	210,212	092	i	90	4s	0	36.5	11.9	54.5	Tagline Broke
213	22-Jul-97	TASCA3	212,214	093	i	90	2	90	36.5	11.9	54.5	
215	22-Jul-97	TASCA4	214,216	094	i	0	3	90	36.5	11.9	54.5	
217	22-Jul-97	TASCA4	216	095	i	90	3	90	36.5	11.9	54.5	
219	22-Jul-97	TASCA4	218,220	096	i	45	2	0	36.5	11.9	54.5	
221	22-Jul-97	TASCA4	220,222	097	i	315	2	0	36.5	11.9	54.5	
223	22-Jul-97	TASCA4	222,224	098	i	45	3	0	36.5	11.9	54.5	
225	22-Jul-97	TASCA4	224,226	099	i	315	3	0	36.5	11.9	54.5	
227	22-Jul-97	TASCA4	226,228	100	i	45	4	0	36.5	11.9	54.5	
229	22-Jul-97	TASCA4	228,230	101	i	315	4	0	36.5	11.9	54.5	
231	22-Jul-97	TASCA4	230,232	102	i	45	3s	0	36.5	11.9	54.5	
233	22-Jul-97	TASCA4	232,234	103	i	315	3s	0	36.5	11.9	54.5	
235	22-	TASCA4	234,236	104	i	45	4s	0	36.5	11.9	54.5	

	Jul- 97											
237	22- Jul- 97	TASCA4	236,238	105	i	315	4s	0	36.5	11.9	54.5	
240	22- Jul- 97	TASCA4	239,241	106	i	45	2	0	36.5	5.5	54.5	
242	22- Jul- 97	TASCA4	241,243	107	i	315	2	0	36.5	5.5	54.5	
249	23- Jul- 97	TASCA4	248,250	108	i	45	3	0	36.5	5.5	54.5	
251	23- Jul- 97	TASCA4	250,252	109	i	315	3	0	36.5	5.5	54.5	
253	23- Jul- 97	TASCA4	252,254	110	i	45	3s	0	36.5	5.5	54.5	
255	23- Jul- 97	TASCA4	254,256	111	i	315	3s	0	36.5	5.5	54.5	24 File Format
257	23- Jul- 97	TASCA4	256,258	112	i	45	4	0	36.5	5.5	54.5	
259	23- Jul- 97	TASCA4	258,260	113	i	315	4	0	36.5	5.5	54.5	
261	23- Jul- 97	TASCA4	260,262	114	i	45	4s	0	36.5	5.5	54.5	
263	23- Jul- 97	TASCA4	262,264	115	i	315	4s	0	36.5	5.5	54.5	
265	23- Jul- 97	TASCA4	264,266	116	i	45	2	0	26.7	5.5	54.5	
268	23- Jul- 97	TASCA4	267,269	117	i	315	2	0	26.7	5.5	54.5	
270	23- Jul- 97	TASCA4	269,271	118	i	45	3	0	26.7	5.5	54.5	
272	23- Jul- 97	TASCA4	271,273	119	i	315	3	0	26.7	5.5	54.5	
274	23- Jul- 97	TASCA4	273,275	120	i	45	3s	0	26.7	5.5	54.5	
276	23- Jul- 97	TASCA4	275,277	121	i	315	3s	0	26.7	5.5	54.5	NRL Data Lost.
278	23- Jul- 97	TASCA4	277,279	122	i	45	4	0	26.7	5.5	54.5	
280	23- Jul- 97	TASCA4	279,281	123	i	315	4	4	26.7	5.5	54.5	
283	23- Jul- 97	TASCA4	282,284	124	i	45	2	0	0	0	54.5	
285	23- Jul-	TASCA4	284,286	125	i	315	2	0	0	0	54.5	

	97											
287	23-Jul-97	TASCA4	286,288	126	i	45	3	0	0	0	54.5	
289	23-Jul-97	TASCA4	288,290	127	i	315	3	0	0	0	54.5	
291	23-Jul-97	TASCA4	290,292	128	i	45	3s	0	0	0	54.5	
293	23-Jul-97	TASCA4	292,294	129	i	315	3s	0	0	0	54.5	
296	23-Jul-97	TASCA4	295,297	130	i	45	2	270	0	0	54.5	
298	23-Jul-97	TASCA4	297,299	131	i	315	2	270	0	0	54.5	
300	23-Jul-97	TASCA4	299,301	132	i	45	3	270	0	0	54.5	
308	24-Jul-97	TASCA4	307,309	133	i	315	3	270	0	0	54.5	
310	24-Jul-97	TASCA4	309,311	134	i	45	3s	270	0	0	54.5	
312	24-Jul-97	TASCA4	311,313	135	i	315	3s	270	0	0	54.5	
316	24-Jul-97	TASCA5	315,317	136	i	45	2	270	36.5	5.5	54.5	
318	24-Jul-97	TASCA5	317,319	137	i	315	2	270	36.5	5.5	54.5	
320	24-Jul-97	TASCA5	319,321	138	i	45	3	270	36.5	5.5	54.5	
322	24-Jul-97	TASCA5	321,323	139	i	315	3	270	36.5	5.5	54.5	
324	24-Jul-97	TASCA5	323,325	140	i	45	3s	270	36.5	5.5	54.5	
326	24-Jul-97	TASCA5	325,327	141	i	315	3s	270	36.5	5.5	54.5	
328	24-Jul-97	TASCA5	327,329	142	i	45	4	270	36.5	5.5	54.5	
330	24-Jul-97	TASCA5	329,331	143	i	315	4	270	36.5	5.5	54.5	
332	24-Jul-97	TASCA5	331,333	144	i	45	4s	270	36.5	5.5	54.5	
334	24-Jul-97	TASCA5	333,335	145	i	315	4s	270	36.5	5.5	54.5	
340	24-Jul-97	TASCA5	339,341	146	i	45	2	270	36.5	12	54.5	

342	24-Jul-97	TASCA5	341,343	147	i	315	2	270	36.5	12	54.5	
344	24-Jul-97	TASCA5	343,345	148	i	45	3	270	36.5	12	54.5	
346	24-Jul-97	TASCA5	345,347	149	i	315	3	270	36.5	12	54.5	
348	24-Jul-97	TASCA5	347,349	150	i	45	3s	270	36.5	12	54.5	
350	24-Jul-97	TASCA5	349,351	151	i	315	3s	270	36.5	12	54.5	
352	24-Jul-97	TASCA5	351,353	152	i	45	4	270	36.5	12	54.5	
354	24-Jul-97	TASCA5	353,355	153	i	315	4	270	36.5	12	54.5	
356	24-Jul-97	TASCA5	355,357	154	i	45	4s	270	36.5	12	54.5	
358	24-Jul-97	TASCA5	357,359	155	i	315	4s	270	36.5	12	54.5	
364	24-Jul-97	TASCA5	363,365	156	i	45	2	270	36.5	12	29.6	
366	24-Jul-97	TASCA5	365,367	157	i	315	2	270	36.5	12	29.6	
368	24-Jul-97	TASCA5	367,369	158	i	45	3	270	36.5	12	29.6	
370	24-Jul-97	TASCA5	369,371	159	i	315	3	270	36.5	12	29.6	
372	24-Jul-97	TASCA5	371,373	160	i	45	4	270	36.5	12	29.6	
374	24-Jul-97	TASCA5	373	161	i	315	4	270	36.5	12	29.6	
380	25-Jul-97	TASCA5	379,381	162	i	45	3s	270	36.5	12	29.6	
382	25-Jul-97	TASCA5	381,383	163	i	315	3s	270	36.5	12	29.6	
384	25-Jul-97	TASCA5	383,385	164	i	45	4s	270	36.5	12	29.6	
386	25-Jul-97	TASCA5	385,387	165	i	315	4s	270	36.5	12	29.6	
390	25-Jul-97	TASCA5	389,391	166	i	270	2	270	36.5	12	29.6	
392	25-Jul-97	TASCA5	391,393	167	i	270	3	270	36.5	12	29.6	
394	25-	TASCA5	393,395	168	i	180	3	270	36.5	12	29.6	

	Jul-97											
396	25-Jul-97	TASCA5	395,397	169	i	270	3s	270	36.5	12	29.6	
398	25-Jul-97	TASCA5	397,399	170	i	180	3s	270	36.5	12	29.6	24 file format.
401	25-Jul-97	TASCA5	400,402	171	i	270	2	270	36.5	12	54.5	
403	25-Jul-97	TASCA5	402,404	172	i	270	3	270	36.5	12	54.5	
405	25-Jul-97	TASCA5	404,406	173	i	270	3s	270	36.5	12	54.5	
407	25-Jul-97	TASCA5	406,408	174	i	270	4	270	36.5	12	54.5	
410	25-Jul-97	TASCA5	409,411	175	i	270	2	270	36.5	5.5	54.5	
412	25-Jul-97	TASCA5	411,413	176	i	270	3	270	36.5	5.5	54.5	
414	25-Jul-97	TASCA5	413,415	177	i	270	3s	270	36.5	5.5	54.5	
417	25-Jul-97	TASCA6	416,418	178	i	270	2	0	36.5	5.5	54.5	
419	25-Jul-97	TASCA6	418,420	179	i	270	3	0	36.5	5.5	54.5	
421	25-Jul-97	TASCA6	420,422	180	i	270	3s	0	36.5	5.5	54.5	
423	25-Jul-97	TASCA6	422,424	181	i	270	3s	0	36.5	12	54.5	NRL first 9 chans of lighter accel only.
425	25-Jul-97	TASCA6	424,426	182	i	270	3	0	36.5	12	54.5	
430	14-Aug-97	TASCA7	429,431	183	i	180	2	270	36.5	12	54.5	
432	14-Aug-97	TASCA7	431,433	184	I	180	3	270	36.5	12	54.5	
434	14-Aug-97	TASCA7	433,435	185	I	270	3	270	36.5	12	54.5	No match between DTMB and NRL data.
436	14-Aug-97	TASCA7	435,437	186	I	180	3s	270	36.5	12	54.5	
438	14-Aug-97	TASCA7	437,439	187	I	180	4	270	36.5	12	54.5	
444	14-Aug-97	TASCA7	443,445	188	I	180	4+	270	36.5	12	54.5	
446	14-Aug-	TASCA7	445,447	189	I	180	4s	270	36.5	12	54.5	24 File format

	97											
448	14-Aug-97	TASCA7	447,449	190	I	180	2	270	0	0	54.5	
450	14-Aug-97	TASCA7	449,451	191	I	180	3	270	0	0	54.5	
452	14-Aug-97	TASCA7	451,453	192	I	180	3s	270	0	0	54.5	
454	14-Aug-97	TASCA7	453,455	193	I	180	4	270	0	0	54.5	
456	14-Aug-97	TASCA7	455,457	194	I	180	4+	270	0	0	54.5	
458	14-Aug-97	TASCA7	457	195	I	180	4s	270	0	0	54.5	
460	15-Aug-97	TASCA7	459,461	196	I	180	2	315	0	0	25	
462	15-Aug-97	TASCA7	461,463	197	I	270	2	315	0	0	25	
464	15-Aug-97	TASCA7	463,465	198	I	180	3	315	0	0	25	
466	15-Aug-97	TASCA7	465,467	199	I	270	3	315	0	0	25	
468	15-Aug-97	TASCA7	467,469	200	I	180	3s	315	0	0	25	
470	15-Aug-97	TASCA7	469,471	201	I	270	3s	315	0	0	25	
472	15-Aug-97	TASCA7	471,473	202	I	180	4	315	0	0	25	
474	15-Aug-97	TASCA7	473,475	203	I	270	4	315	0	0	25	NRL Data Lost.
476	15-Aug-97	TASCA7	475,477	204	I	180	4+	315	0	0	25	NRL Data Lost.
478	15-Aug-97	TASCA7	477,479	205	I	180	4s	315	0	0	25	
480	15-Aug-97	TASCA7	479,481	206	I	270	4s	315	0	0	25	
482	15-Aug-97	TASCA7	481,483	207	I	270	4+	315	0	0	25	
484	15-Aug-97	TASCA7	483,485	208	I	180	2	315	29.25	12	25	
486	15-Aug-97	TASCA7	485,487	209	I	270	2	315	29.25	12	25	
488	15-Aug-97	TASCA7	487,489	210	I	180	3	315	29.25	12	25	NRL Data Lost.

490	15-Aug-97	TASCA7	489,491	211	I	270	3	315	29.25	12	25	NRL Format 2.
492	15-Aug-97	TASCA7	491,493	212	I	180	3s	315	29.25	12	25	
494	15-Aug-97	TASCA7	493,495	213	I	270	3s	315	29.25	12	25	
496	15-Aug-97	TASCA7	495,497	214	I	180	4	315	29.25	12	25	NRL Format 2.
498	15-Aug-97	TASCA7	497,499	215	I	270	4	315	29.25	12	25	NRL Format 2.
500	15-Aug-97	TASCA7	499,501	216	I	180	4s	315	29.25	12	25	NRL Format 2.
502	15-Aug-97	TASCA7	501	217	I	270	4s	315	29.25	12	25	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TEST CONFIGURATION II

CDNSWC Run #	Date	Run Storage Directory	Reference Zero Run	NRL Run #	Test Config #	Wave Heading	Sea State	Boom Slew Angle	Rider Block Location	Rider Block Inhaul Angle	Boom Luff Angle	Comments
504	15-Aug-97	TACSA7	503,505	218	Ii	180	3	315	5	12	25	
506	15-Aug-97	TACSA7	505,507	219	Ii	270	3	315	5	12	25	
508	15-Aug-97	TACSA7	507,509	220	Ii	270	3s	315	5	12	25	
510	15-Aug-97	TACSA7	509,511	221	Ii	270	4	315	5	12	25	
512	15-Aug-97	TACSA7	511,513	222	Ii	270	4s	315	5	12	25	
515	16-Aug-97	TACSA7	514,516	223 *	Ii	90	3	270	5	12	25	NRL Load Data Not Locked
517	16-Aug-97	TACSA7	516,518	224	Ii	90	3s	270	5	12	25	NRL Format 2.
519	16-Aug-97	TACSA7	518,520	225	Ii	90	4	270	5	12	25	
521	16-Aug-97	TACSA7	520,522	226	Ii	90	4s	270	5	12	25	
523	16-Aug-97	TACSA7	522,524	223	Ii	90	3	270	5	12	25	Rerun to get good load data.
526	16-Aug-97	TACSA7	525	NCD	Ii	90	4					Load Control Configuration 1
527	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Load Control Configuration 2
528	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Load Control Configuration 2
529	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Transition Control Configuration 2 to 3
530	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Load Control Configuration 3
531	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Dynamic Load Control Demonstration
532	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Load Control Configuration 4
533	16-Aug-97	TASCA8	525	NCD	Ii	90	4					Dynamic Load Demo No Rider Block

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: TEST CONFIGURATION III

CDNSWC Run #	Date	Run Storage Directory	Reference Zero Run	NRL Run #	Test Config #	Wave Heading	Sea State	Boom Slew Angle	Rider Block Location	Rider Block Inhaul Angle	Boom Luff Angle	Comments
537	17-Aug-97	TASCA9	536,538	NCD	Iii	0	2	45	9.5	16	60	
539	17-Aug-97	TASCA9	538,540	227	iii	90	2	45	9.5	16	60	
541	17-Aug-97	TASCA9	540,542	228	iii	0	3	45	9.5	16	60	
543	17-Aug-97	TASCA9	542,544	229	iii	90	3	45	9.5	16	60	
545	17-Aug-97	TASCA9	544,546	230	iii	0	3s	45	9.5	16	60	
547	17-Aug-97	TASCA9	546,548	231	iii	90	3s	45	9.5	16	60	
549	17-Aug-97	TASCA9	548,550	232	iii	0	4	45	9.5	16	60	
551	17-Aug-97	TASCA9	550,552	233	iii	90	4	45	9.5	16	60	
553	17-Aug-97	TASCA9	552,554	234	iii	0	4+	45	9.5	16	60	
555	17-Aug-97	TASCA9	554,556	235	iii	0	4s	45	9.5	16	60	
557	17-Aug-97	TASCA9	556,558	NCD	iii	90	3	45	5.625	15	60	
559	17-Aug-97	TASCA9	558,560	236	iii	0	2	45	39.5	0	60	
561	17-Aug-97	TASCA9	560,562	237	iii	90	2	45	39.5	0	60	
563	17-Aug-97	TASCA9	562,564	238	iii	0	3	45	39.5	0	60	
565	17-Aug-97	TASCA9	564,566	239	iii	90	3	45	39.5	0	60	
568	17-Aug-97	TASCA9	567,569	240	iii	0	2	90	45.3	13	60	
570	17-Aug-97	TASCA9	569,571	241	iii	90	2	90	45.3	13	60	
572	17-Aug-97	TASCA9	571,573	242	iii	0	3	90	45.3	13	60	
574	17-	TASCA9	573,575	243	iii	90	3	90	45.3	13	60	

	Aug-97											
576	17-Aug-97	TASCA9	575,577	244	iii	0	3s	90	45.3	13	60	
578	17-Aug-97	TASCA9	577,579	245	iii	90	3s	90	45.3	13	60	
580	17-Aug-97	TASCA9	579,582	246	iii	0	4	90	45.3	13	60	
581	17-Aug-97	TASCA9	579,582	247	iii	0	4+	90	45.3	13	60	
583	17-Aug-97	TASCA9	582	248	iii	90	3	90	45.3	13	60	240secs Static + Load Control Demo
584	17-Aug-97	TASCA9	582	249	iii	0	2	90	0	0	60	240secs Static + Load Control Demo
585	17-Aug-97	TASCA9	582	250	iii	90	2	90	0	0	60	240secs Static + Load Control Demo
586	17-Aug-97	TASCA9	582	251	iii	90	3	90	0	0	60	200secs Static + Load Control Demo

APPENDIX D: MASTER CHANNEL LIST: CONFIGURATION I

DTMB Runs 1-503 and NRL Runs 1-217

Channel #	Channel Title	Units	Final File Title
01	Wave Ht Bow	inch	r###ch01
02	Sonix Sonic	inch	r###ch02
03	TACS Roll	degree	r###ch03
04	TACS Pitch	degree	r###ch04
05	TACS Roll Rt	deg/sec	r###ch05
06	TACS PitchRt	deg/sec	r###ch06
07	TACS Yaw Rt	deg/sec	r###ch07
08	TACS CG ZAcc	G	r###ch08
09	TACS CG YAcc	G	r###ch09
10	TACS CG XAcc	G	r###ch10
11	TACS CT ZAcc	G	r###ch11
12	TACS CT YAcc	G	r###ch12
13	Cntr Roll	degree	r###ch13
14	Cntr Pitch	degree	r###ch14
15	Cntr CG ZAcc	G	r###ch15
16	Cntr CG YAcc	G	r###ch16
17	Cntr CG XAcc	G	r###ch17
18	TACS Cntr Rel X	inch	r###ch18
19	TACS Cntr Rel Y	inch	r###ch19
20	TACS Cntr Rel Z	inch	r###ch20
21	Lhtr Roll	degree	r###ch21
22	Lhtr Pitch	degree	r###ch22
23	Lhtr Roll Rt	deg/sec	r###ch23
24	Lhtr PitchRt	deg/sec	r###ch24
25	Lhtr CG ZAcc	G	r###ch25
26	Lhtr CG YAcc	G	r###ch26
27	Lhtr CG XAcc	G	r###ch27
28	TACS Lhtr Rel X	inch	r###ch28
29	TACS Lhtr Rel Y	inch	r###ch29
30	TACS Lhtr Rel Z	inch	r###ch30
31	Load Mo Orthognl	inch	r###ch31
32	Load Mo Parallel	inch	r###ch32
33	BoomTip-HorizAcc	G	r###ch33
34	BoomTip-Vert Acc	G	r###ch34
35	Lghtr4-Strn VAcc	G	r###ch35
36	Lghtr4-Bow VAcc	G	r###ch36
37	Lghtr3-Bow VAcc	G	r###ch37

38	Lghtr2-Bow VAcc	G	r###ch38
39	Lghtr1-Bow VAcc	G	r###ch39
40	Lghtr2-SMdShpVAc	G	r###ch40
41	Lghtr2-PMdShpVAc	G	r###ch41
42	Lghtr2-CLMdSpVAc	G	r###ch42
43	Lghtr1-CLMdSpVAc	G	r###ch43
44	Lghtr4-StrnLngAc	G	r###ch44
45	Lghtr4-PStrnTvAc	G	r###ch45
46	Lghtr1-PBowTvAcc	G	r###ch46

APPENDIX E. MASTER CHANNEL LIST: CONFIGURATION II

DTMB Runs 504-533 and NRL Runs 218-226

Channel #	Channel Title	Units	Final File Title
01	Wave Ht Bow	inch	r###ch01
02	Sonix Sonic	inch	r###ch02
03	TACS Roll	degree	r###ch03
04	TACS Pitch	degree	r###ch04
05	TACS Roll Rt	deg/sec	r###ch05
06	TACS PitchRt	deg/sec	r###ch06
07	TACS Yaw Rt	deg/sec	r###ch07
08	TACS CG ZAcc	G	r###ch08
09	TACS CG YAcc	G	r###ch09
10	TACS CG XAcc	G	r###ch10
11	TACS CT ZAcc	G	r###ch11
12	TACS CT YAcc	G	r###ch12
13	Lhtr Roll	degree	r###ch13
14	Lhtr Pitch	degree	r###ch14
15	Lntr Roll Rt	deg/sec	r###ch15
16	Lhtr PitchRt	deg/sec	r###ch16
17	Lhtr CG ZAcc	G	r###ch17
18	Lhtr CG YAcc	G	r###ch18
19	Lhtr CG XAcc	G	r###ch19
20	TACS LhtRelX	inch	r###ch20
21	TACS LhtRelY	inch	r###ch21
22	TACS LhtRelZ	inch	r###ch22
23	Load Mo Orthognl	inch	r###ch23
24	Load Mo Parallel	inch	r###ch24
25	BoomTip-HorizAcc	G	r###ch25
26	BoomTip-Vert Acc	G	r###ch26
27	Lghtr4-Strn VAcc	G	r###ch27
28	Lghtr4-Bow VAcc	G	r###ch28
29	Lghtr3-Bow VAcc	G	r###ch29
30	Lghtr2-Bow VAcc	G	r###ch30
31	Lghtr1-Bow VAcc	G	r###ch31
32	Lghtr2-SMdShpVAc	G	r###ch32
33	Lghtr2-PMdShpVAc	G	r###ch33
34	Lghtr2-CLMdSpVAc	G	r###ch34
35	Lghtr1-CLMdSpVAc	G	r###ch35
36	Lghtr4-StrnLngAc	G	r###ch36
37	Lghtr4-PStrnTAcc	G	r###ch37
38	Lghtr1-PBowTvAcc	G	r###ch38

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. MASTER CHANNEL LIST: CONFIGURATION III

DTMB Runs 534-587 and NRL Runs 227-251

Channel #	Channel Title	Units	Final File Title
01	Wave Ht Bow	inch	r###ch01
02	Sonix Sonic	inch	r###ch02
03	TACS Roll	degree	r###ch03
04	TACS Pitch	degree	r###ch04
05	TACS Roll Rt	deg/sec	r###ch05
06	TACS PitchRt	deg/sec	r###ch06
07	TACS Yaw Rt	deg/sec	r###ch07
08	TACS CG ZAcc	G	r###ch08
09	TACS CG YAcc	G	r###ch09
10	TACS CG XAcc	G	r###ch10
11	TACS CT ZAcc	G	r###ch11
12	TACS CT YAcc	G	r###ch12
13	DD 963 Roll	degree	r###ch13
14	DD 963 Pitch	degree	r###ch14
15	DD 963 CG ZAcc	G	r###ch15
16	DD 963 CG YAcc	G	r###ch16
17	DD 963 CG XAcc	G	r###ch17
18	TACS1 DD963 RelX	inch	r###ch18
19	TACS1 DD963 RelY	inch	r###ch19
20	TACS1 DD963 RelZ	Inch	r###ch20
21	Load Mo Orthognl	Inch	r###ch21
22	Load Mo Parallel	Inch	r###ch22
23	BoomTip-HorizAcc	G	r###ch23
24	BoomTip-Vert Acc	G	r###ch24

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. RAO PROGRAM

```
function rao(run)

% Program Title: rao.m-------------------------------------
% Referred to as RAO Program

% Must enter one of the following to initiate this program:
%
%    rao(##)    for DTMB run numbers <100
%    rao(###)     for DTMB run numbers >100

% Initial Set-up----------------------------------------------
sread           = char('H:\');
swrite          = char('D:\');

format short e

ptavg           = 50;
window          = 500;
lambda          = 24.175;
freq            = 32.2;

% Establish Proper String Configuration for Data Run-------
if run <= 99
     s1         = char('R0');
     s2         = char('0');
   else
     s1         = char('R');
end

srun            = num2str(run);
filename        = strcat(s1,srun);

if run <= 99
     folder     = strcat(s2,srun);
    else
      folder    = srun;
end
```

```
% Determine Number of Data Channels to be Processed--------
if run <= 503
      channels  =  46;
   elseif run <= 533
       channels =  38;
   else
     channels  =  24;
end

% Establish Filepath and Load the Raw, Time-based Data-----
filepath        =  [sread,filename];
load(filepath)
szdata          =  size(data);
channels        =  szdata(2);
n               =  szdata(1);

% Compute both Model-Scale and Full-Scale Omega Ranges-----
momega          = 2*pi*freq/n*(0:window-1);
fomega          = momega/sqrt(lambda);

% Full-Scale Scalers, PSD Units and RAO Units Library------
fs1             = 1;
pu1             = '(in)';
ru1             = '(dimensionless)';

fs2             = 12/lambda;
pu2             = '(deg)';
ru2             = '(deg/in)';

fs3             = 12/lambda^(3/2);
pu3             = '(deg/s)';
ru3             = '(deg/in s)';

fs4             = 12/lambda^2;
pu4             = '(deg/s^2)';
ru4             = '(deg/in s^2)';

% Looping/Processing of Data Channels---------------------
for ch = 1:channels

   if channels==46

   if ch==1
        fscaler   = fs1;
        psdunits  = pu1;
        raounits  = ru1;
     chtitle   = ’Wave Ht Bow’;
     elseif ch==2
          fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = ’Sonix Sonic’;
elseif ch==3
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = ’TACS Roll’;
elseif ch==4
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = ’TACS Pitch’;
elseif ch==5
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = ’TACS Roll Rt’;
elseif ch==6
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = ’TACS Pitch Rt’;
elseif ch==7
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = ’TACS Yaw Rt’;
elseif ch==8
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’TACS CG ZAcc’;
elseif ch==9
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’TACS CG YAcc’;
elseif ch==10
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’TACS CG XAcc’;
```

```
elseif ch==11
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CT ZAcc';
elseif ch==12
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CF YAcc';
elseif ch==13
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'Cntr Roll';
elseif ch==14
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'Cntr Pitch';
elseif ch==15
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Cntr CG ZAcc';
elseif ch==16
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Cntr CG YAcc';
elseif ch==17
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Cntr CG XAcc';
elseif ch==18
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS Cntr Rel X';
elseif ch==19
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS Cntr Rel Y';
```

```
elseif ch==20
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS Cntr Rel Z';
elseif ch==21
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'Lhtr Roll';
elseif ch==22
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'Lhtr Pitch';
elseif ch==23
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'Lhtr Roll Rt';
elseif ch==24
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'Lhtr Pitch Rt';
elseif ch==25
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lhtr CG ZAcc';
elseif ch==26
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lhtr CG YAcc';
elseif ch==27
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lhtr CG XAcc';
elseif ch==28
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS Lhtr Rel X';
```

```
elseif ch==29
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS Lhtr Rel Y';
elseif ch==30
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS Lhtr Rel Z';
elseif ch==31
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'Load Mo Orhognl';
elseif ch==32
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'Load Mo Parallel';
elseif ch==33
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Boom Tip-HorizAcc';
elseif ch==34
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Boom Tip-VertAcc';
elseif ch==35
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr4-Strn VAcc';
elseif ch==36
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr4-Bow VAcc';
elseif ch==37
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr3-Bow VAcc';
```

```
elseif ch==38
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr2-Bow VAcc';
elseif ch==39
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr1-Bow VAcc';
elseif ch==40
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr2-SMdShp VAcc';
elseif ch==41
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr2-PMdShp VAcc';
elseif ch==42
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr2-CLMdSp VAcc';
elseif ch==43
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr1-CLMdSp VAcc';
elseif ch==44
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr4-StrnLng Acc';
elseif ch==45
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr4-PStrn TvAcc';
else
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lghtr1-PBow TvAcc';
end  % End of Configuration I sub-Loop
```

```
elseif channels==38

if ch==1
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'Wave Ht Bow';
elseif ch==2
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'Sonix Sonic';
elseif ch==3
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'TACS Roll';
elseif ch==4
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'TACS Pitch';
elseif ch==5
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'TACS Roll Rt';
elseif ch==6
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'TACS Pitch Rt';
elseif ch==7
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'TACS Yaw Rt';
elseif ch==8
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CG ZAcc';
```

```
elseif ch==9
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CG YAcc';
elseif ch==10
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CG XAcc';
elseif ch==11
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CT ZAcc';
elseif ch==12
     fsscaler    = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'TACS CF YAcc';
elseif ch==13
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'Lhtr Roll';
elseif ch==14
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = 'Lhtr Pitch';
elseif ch==15
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'Lhtr Roll Rt';
elseif ch==16
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = 'Lhtr Pitch RT';
elseif ch==17
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'Lhtr CG ZAcc';
```

```
elseif ch==18
     fscaler   =  fs4;
     psdunits  =  pu4;
     raounits  =  ru4;
     chtitle   =  'Lhtr CG YAcc';
elseif ch==19
     fscaler   =  fs4;
     psdunits  =  pu4;
     raounits  =  ru4;
     chtitle   =  'Lhtr CG XAcc';
elseif ch==20
     fscaler   =  fs1;
     psdunits  =  pu1;
     raounits  =  ru1;
     chtitle   =  'TACS LhtRelX';
elseif ch==21
     fscaler   =  fs1;
     psdunits  =  pu1;
     raounits  =  ru1;
     chtitle   =  'TACS LhtRelY';
elseif ch==22
     fscaler   =  fs1;
     psdunits  =  pu1;
     raounits  =  ru1;
     chtitle   =  'TACS LhtRelZ';
elseif ch==23
     fscaler   =  fs1;
     psdunits  =  pu1;
     raounits  =  ru1;
     chtitle   =  'Load Mo Orthognl';
elseif ch==24
     fscaler   =  fs1;
     psdunits  =  pu1;
     raounits  =  ru1;
     chtitle   =  'Load Mo Parallel';
elseif ch==25
     fscaler   =  fs4;
     psdunits  =  pu4;
     raounits  =  ru4;
     chtitle   =  'BoomTip-HorizAcc';
elseif ch==26
     fscaler   =  fs4;
     psdunits  =  pu4;
     raounits  =  ru4;
     chtitle   =  'BoomTip-Vert Acc';
```

```
elseif ch==27
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr4-Strn VAcc’;
elseif ch==28
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr4-Bow VAcc’;
elseif ch==29
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr3-Bow VAcc’;
elseif ch==30
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr2-Bow VAcc’;
elseif ch==31
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr1-Bow VAcc’;
elseif ch==32
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr2-SMdShpVAc’;
elseif ch==33
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr2-PMdShpVAc’;
elseif ch==34
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr2-CLMdSpVAc’;
elseif ch==35
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr1-CLMdSpVAc’;
```

```
elseif ch==36
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr4-StrnLngAc’;
elseif ch==37
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr4-PStrnTAcc’;
else
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = ’Lghtr1-PBowTvAcc’;
end  % End of Configuration II sub-Loop

else

     if ch==1
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = ’Wave Ht Bow’;
elseif ch==2
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = ’Sonix Sonic’;
     elseif ch==3
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = ’TACS Roll’;
     elseif ch==4
     fscaler   = fs2;
     psdunits  = pu2;
     raounits  = ru2;
     chtitle   = ’TACS Pitch’;
     elseif ch==5
     fscaler   = fs3;
     psdunits  = pu3;
     raounits  = ru3;
     chtitle   = ’TACS Roll Rt’;
```

```
elseif ch==6
fscaler   =  fs3;
psdunits  =  pu3;
raounits  =  ru3;
chtitle   =  'TACS Pitch Rt';
elseif ch==7
fscaler   =  fs3;
psdunits  =  pu3;
raounits  =  ru3;
chtitle   =  'TACS Yaw Rt';
elseif ch==8
fscaler   =  fs4;
psdunits  =  pu4;
raounits  =  ru4;
chtitle   =  'TACS CG ZAcc';
elseif ch==9
fscaler   =  fs4;
psdunits  =  pu4;
raounits  =  ru4;
chtitle   =  'TACS CG YAcc';
elseif ch==10
fscaler   =  fs4;
psdunits  =  pu4;
raounits  =  ru4;
chtitle   =  'TACS CG XAcc';
elseif ch==11
fscaler   =  fs4;
psdunits  =  pu4;
raounits  =  ru4;
chtitle   =  'TACS CT ZAcc';
elseif ch==12
fscaler   =  fs4;
psdunits  =  pu4;
raounits  =  ru4;
chtitle   =  'TACS CF YAcc';
elseif ch==13
fscaler   =  fs2;
psdunits  =  pu2;
raounits  =  ru2;
chtitle   =  'DD 963 Roll';
elseif ch==14
fscaler   =  fs2;
psdunits  =  pu2;
raounits  =  ru2;
chtitle   =  'DD 963 Pitch';
```

```
     elseif ch==15
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'DD 963 CG ZAcc';
     elseif ch==16
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'DD 963 CG YAcc';
     elseif ch==17
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'DD 963 CG XAcc';
     elseif ch==18
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS1 DD963 RelX';
elseif ch==19
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS1 DD963 RelY';
     elseif ch==20
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'TACS1 DD963 RelZ';
     elseif ch==21
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'Load Mo Orthognl';
     elseif ch==22
     fscaler   = fs1;
     psdunits  = pu1;
     raounits  = ru1;
     chtitle   = 'Load Mo Parallel';
     elseif ch==23
     fscaler   = fs4;
     psdunits  = pu4;
     raounits  = ru4;
     chtitle   = 'BoomTip-HorizAcc';
```

```
        else
        fscaler    =  fs4;
        psdunits   =  pu4;
        raounits   =  ru4;
        chtitle    =  ’BoomTip-Vert Acc’;
  end   % End of Configuration III sub-Loop

  end   % End of the Overall Configuration Loop

% Base-lining the Raw, Time-based Data---------------------
  x                =  data(:,ch);
  x                =  x-mean(x);

% FFT Computation------------------------------------------
  Y                =  fft(x,n);

% Model-Scale PSD Computation------------------------------
  PSD              =  Y.*conj(Y)/n;

% Moving-window Averaging of Model-Scale PSDs for
  smoothing------------------------------------------------
  halfptavg        =  ptavg/2;
  sum              =  0;
  for k = 1:ptavg
       sum         =  sum + PSD(k,1);
       if k > halfptavg
            psd(k-halfptavg,ch) =  sum/k;
          else
       end
  end
  pts              =  window + halfptavg;
  ptavgplusone     =  ptavg + 1;
  j                =  1;
  for k   =  ptavgplusone:pts
     sum           =  sum + PSD(ptavg+j,1) - PSD(j,1);
     psd(k-halfptavg,ch) =  sum/ptavg;
     j             =  j + 1;
  end

  meansqr          =  mean(x.^2);
  psdarea          =  trap(momega,psd(:,ch));
  psdscaler        =  meansqr/psdarea;
  psd(:,ch)        =  psd(:,ch)*psdscaler;
```

```
% Computation of Model-Scale RAOs--------------------------
  X               =  psd(:,ch)./psd(:,1);
  mrao(:,ch)      =  sqrt(X);

% Computation of Full-Scale RAOs---------------------------
  frao(:,ch)      = mrao(:,ch)*fscaler;

% Plot Model-Scale PSDs and Save Figure in jpg Format------
  clf
  plot(momega(1,:),psd(:,ch))
  grid
  sch             = num2str(ch);
  title(['Model-Scale PSDs for DTMB Run 'srun,' / Channel
       ',sch,': 'chtitle])
  xlabel('Omega (rad/s)')
  ylabel(['PSD 'psdunits])

  saveas(gcf,[swrite,folder,'\',filename,'C',sch,'p'],'jpg
       '); % D:\###\R###C##p.jpg

% Plot Model-Scale RAOs and Save Figure in jpg Format------
  clf
  plot(momega(1,:),mrao(:,ch))
  grid
  title(['Model-Scale RAOs for DTMB Run 'srun,' / Channel
       ',sch,': 'chtitle])
  xlabel('Omega (rad/s)')
  ylabel(['RAO 'raounits])

  saveas(gcf,[swrite,folder,'\',filename,'C',sch,'m'],'jpg
       '); % D:\###\R###C##m.jpg

% Plot Full-Scale RAOs and Save Figure in jpg Format-------
  clf
  plot(fomega(1,:),frao(:,ch))
  grid
  title(['Full-Scale RAOs for DTMB Run 'srun,' / Channel
       ',sch,': 'chtitle])
  xlabel('Omega (rad/s)')
  ylabel(['RAO 'raounits])

  saveas(gcf,[swrite,folder,'\',filename,'C',sch,'f'],'jpg
       '); % D:\###\R###C##f.jpg

end          % End of all Channel Looping
```

```
%  Save Omega Matrices as Last Column of Composite PSD and
   RAO Matrices---------------------------------------------
ch                  =  ch + 1;
psd(:,ch)           =  momega(1,:)’;
mrao(:,ch)          =  momega(1,:)’;
frao(:,ch)          =  fomega(1,:)’;

%  Save Composite PSD and RAO Matrices----------------------
%      D:\###\R###psd.txt
%      D:\###\R###mrao.txt
%      D:\###\R###frao.txt
save([swrite,folder,’\’,filename,’psd.txt’],’psd’,’-ascii
     ’,’-tabs’);
save([swrite,folder,’\’,filename,’mrao.txt’],’mrao’,’-ascii
     ’,’-tabs’);
save([swrite,folder,’\’,filename,’frao.txt’],’frao’,’-ascii
     ’,’-tabs’);

%  End of RAO Program---------------------------------------
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H. PTAVG COMPARISON PROGRAM

```
% Program title: ptavgcomparison.m------------------------
clear
clc

% Load Spreadsheet Results for run 67---------------------
load run67.txt;

% Load RAO Program Results for ptavg = 26, 50, 76 and 100--
load R067psd26.txt;
load R067psd50.txt;
load R067psd76.txt;
load R067psd100.txt;

load R067mrao26.txt;
load R067mrao50.txt;
load R067mrao76.txt;
load R067mrao100.txt;

load R067frao26.txt;
load R067frao50.txt;
load R067frao76.txt;
load R067frao100.txt;

% Plot Model-Scale PSD Comparison------------------------
figure(1)
plot(R067psd26(:,47),R067psd26(:,3),...
   R067psd50(:,47),R067psd50(:,3),...
   R067psd76(:,47),R067psd76(:,3),...
   R067psd100(:,47),R067psd100(:,3),...
run67(:,2),run67(:,4))
title('R067CH3 Model-Scale TACS Roll PSDs "ptavg"
     Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (deg)')
grid
```

```
% Plot Model-Scale RAO Comparison---------------------------
figure(2)
plot(R067mrao26(:,47),R067mrao26(:,3),...
     R067mrao50(:,47),R067mrao50(:,3),...
     R067mrao76(:,47),R067mrao76(:,3),...
     R067mrao100(:,47),R067mrao100(:,3),...
run67(:,2),run67(:,5))
title('R067CH3 Model-Scale TACS Roll RAOs "ptavg"
     Comparison')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
grid

% Plot Full-Scale RAO Comparison----------------------------
figure(3)
plot(R067frao26(:,47),R067frao26(:,3),...
     R067frao50(:,47),R067frao50(:,3),...
     R067frao76(:,47),R067frao76(:,3),...
     R067frao100(:,47),R067frao100(:,3))
title('R067CH3 Full-Scale TACS Roll RAOs "ptavg"
      Comparison')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
grid

% End of ptavgcomparison program---------------------------
```

APPENDIX I. PSD AND RAO COMPARISONS PROGRAM

```
% Program Title:  psdandraocomparisons.m--------------------

% Computational and Graphical Comparison between the
  various Methods-----------------------------------------

clear
clc

format short

% Load Applicable Matrices--------------------------------
load R067;              % Raw, Time-based Data
load run67.txt;         % Spreadsheet PSD and RAO Results
load R067psd.txt;       % RAO Program Model-Scale PSD Results
load R067mrao.txt;      % RAO Program Model-Scale RAO Results
load R067frao.txt;      % RAO Program Full-Scale RAO Results

% Base-line the Raw Time-based----------------------------
x                 = data(:,1);           % Wave Ht Data
x                 = x - mean(x);

z                 = data(:,3);           % TACS Roll Data
z                 = z - mean(z);

% Computation of Wave Ht PSDs and Associated Frequencies---
[hwelch,fwelch]   = pwelch(x,[],[],8096,32.2);
[hperio,fperio]   = periodogram(x,[],8096,32.2);
[hmulti,fmulti]   = pmtm(x,[],8096,32.2);
meansqrx          = mean(x.^2);          % Wave Ht Data
welchareah        = trap(wwelch,hwelch);   % Wave Ht
perioareah        = trap(wperio,hperio);
multiareah        = trap(wmulti,hmulti);
hwelchscaled      = hwelch*meansqrx/welchareah;
hperioscaled      = hperio*meansqrx/perioareah;
hmultiscaled      = hmulti*meansqrx/multiareah;
```

```
% Computation of TACS Roll PSDs and Associated Frequencies-
[rwelch,fwelch]   = pwelch(z,[],[],8096,32.2);
[rperio,fperio]   = periodogram(z,[],8096,32.2);
[rmulti,fmulti]   = pmtm(z,[],8096,32.2);
meansqrz          = mean(z.^2);            % TACS Roll Data
welcharear        = trap(wwelch,rwelch);   % TACS Roll
perioarear        = trap(wperio,rperio);
multiarear        = trap(wmulti,rmulti);
rwelchscaled      = rwelch*meansqrz/welcharear;
rperioscaled      = rperio*meansqrz/perioarear;
rmultiscaled      = rmulti*meansqrz/multiarear;

% Convert Frequency to Omega------------------------------
wwelch            = 2*pi*fwelch;
wperio            = 2*pi*fperio;
wmulti            = 2*pi*fmulti;

% Compute Areas under the Wave Ht PSD Curves--------------
welchareah2       = trap(wwelch,hwelchscaled);
perioareah2       = trap(wperio,hperioscaled);
multiareah2       = trap(wmulti,hmultiscaled);
spreadareah       = trap(run67(:,2),run67(:,3));
raoprogareah      = trap(R067psd(:,47),R067psd(:,1));

% Compare Areas under the Wave Ht PSD Curves against
  meansqrs------------------------------------------------
wavehtpsdareacomparison     = [meansqrx welchareah2 ...
       perioareah2 multiareah2 spreadareah raoprogareah]

% Compute Areas under the TACS Roll PSD Curves------------
welcharear2       = trap(wwelch,rwelchscaled);
perioarear2       = trap(wperio,rperioscaled);
multiarear2       = trap(wmulti,rmultiscaled);
spreadarear       = trap(run67(:,2),run67(:,4));
raoprogarear      = trap(R067psd(:,47),R067psd(:,3));

% Compare Areas under the TACS Roll PSD Curves against
  meansqrs------------------------------------------------

rollpsdareacomparison     = [meansqrz welcharear2 ...
       perioarear2 multiarear2 spreadarear raoprogarear]
```

```
% Graphical Model-Scale Wave Ht PSDs Comparison of All 5
   Methods---------------------------------------------------

figure(1)
plot(wwelch,hwelchscaled)              % Welch Method
hold on
plot(wperio,hperioscaled)              % Periodogram Method
plot(wmulti,hmultiscaled)              % Multi-taper Method
plot(run67(:,2),run67(:,3))            % Spreadsheet Results
plot(R067psd(:,47),R067psd(:,1));      % Rao Program Results

title('R067CH01 Model-Scale Wave Ht PSDs Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (dimesionless)')
axis([0 6 0 .015])
grid

% Graphical Model-Scale Wave Ht PSDs Comparison------------
%    Spreadsheet vs. RAO Program only
figure(2)
plot(run67(:,2),run67(:,3))            % Spreadsheet Results
hold on
plot(R067psd(:,47),R067psd(:,1))       % Rao Program Results

title('R067CH01 Model-Scale Wave Ht PSDs Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (dimesionless)')
axis([0 6 0 .0065])
grid

% Graphical Model-Scale TACS Roll PSDs Comparison of All 5
   Methods---------------------------------------------------
figure(3)
plot(wwelch,rwelchscaled)              % Welch Method
hold on
plot(wperio,rperioscaled)              % Periodogram Method
plot(wmulti,rmultiscaled)              % Multi-taper Method
plot(run67(:,2),run67(:,4))            % Spreadsheet Results
plot(R067psd(:,47),R067psd(:,3));      % Rao Program Results

title('R067CH03 Model-Scale TACS Roll PSDs Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (deg)')
axis([0 6 0 .02])
grid
```

```
% Graphical TACS Roll PSDs Comparison---------------------
%    Spreadsheet vs. RAO Program only
figure(4)
plot(run67(:,2),run67(:,4))          % Spreadsheet Results
hold on
plot(R067psd(:,47),R067psd(:,3))     % Rao Program Results

title('R067CH03 Model-Scale TACS Roll PSDs Comparison')
xlabel('Omega (rad/s)')
ylabel('PSD (deg)')
axis([0 6 0 .009])
grid

% Compute Model-Scale Roll RAOs from MATLAB Methods--------
R1                = rwelchscaled./hwelchscaled;
welchmrao         = sqrt(R1);
R2                = rperioscaled./hperioscaled;
periomrao         = sqrt(R2);
R3                = rmultiscaled./hmultiscaled;
multimrao         = sqrt(R3);

% Graphical Model-Scale TACS Roll RAOs Comparison of All 5
   Methods--------------------------------------------------------
figure(5)
plot(wwelch,welchmrao)               % Welch Method
hold on
plot(wperio,periomrao)               % Periodogram Method
plot(wmulti,multimrao)               % Multi-taper Method
plot(run67(:,2),run67(:,5))          % Spreadsheet Results
plot(R067mrao(:,47),R067mrao(:,3));  % Rao Program Results

title('R067CH03 Model-Scale TACS Roll RAOs Comparison')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
axis([0 6 0 10])
grid

% Graphical Model-Scale TACS Roll RAOs Comparison----------

%    (Spreadsheet vs. RAO Program only)
figure(6)
plot(run67(:,2),run67(:,5))          % Spreadsheet Results
hold on
plot(R067mrao(:,47),R067mrao(:,3))   % Rao Program Results
```

```
title('R067CH03 Model-Scale TACS Roll RAOs Comparison')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
axis([0 6 0 2.2])
grid

% Scale to Full-Scale RAOs and Omegas----------------------
welchfrao         = welchmrao*12/24.175;
periofrao         = periomrao*12/24.175;
multifrao         = multimrao*12/24.175;

wwelchfull        = wwelch/sqrt(24.175);
wperiofull        = wperio/sqrt(24.175);
wmultifull        = wmulti/sqrt(24.175);

% Graphical Full-Scale TACS Roll RAOs Comparison-----------
%    (4 Methods Not including Spreadsheet)
figure(7)
plot(wwelchfull,welchfrao)              % Welch Method
hold on
plot(wperiofull,periofrao)              % Periodogram Method
plot(wmultifull,multifrao)              % Multi-taper Method
plot(R067frao(:,47),R067frao(:,3))      % RAO Program Results

title('R067CH03 Full-Scale TACS Roll RAOs Comparison')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
axis([0 1.4 0 3])
grid

% Graphical Full-Scale TACS Roll RAOs Presentation---------
%    RAO Program only
figure(8)
plot(R067frao(:,47),R067frao(:,3))      % Rao Program Results

title('R067CH03 Full-Scale TACS Roll RAOs Presentation')
xlabel('Omega (rad/s)')
ylabel('RAO (deg/in)')
axis([0 1.4 0 1.4])
grid

% End of PSD and RAO Comparison Program--------------------
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Marple, S.L. Jr., *DIGITAL SPECTRAL ANALYSIS WITH APPLICATIONS*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1987.

Papoulias, F.A., "*Ship Dynamics*", Informal Lecture Notes, Naval Postgraduate School, Monterey, California, Summer 2000.

Signals Processing Toolbox, MATLAB Computer Program: The Language of Technical Computing, ver. 5.3.1.2915a (R11.1), September 28, 1999.

Tupper, E., *Introduction to NAVAL ARCHITECTURE Third Edition*, Butterworth-Heinemann, Oxford, England, 1996.

Zubaly, R.B., *APPLIED NAVAL ARCHITECTURE*, Cornell Maritime Press, Centerville, Maryland, 1996.

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

Bogert, B.P., "Informal Comments on the Uses of Power Spectrum Analysis," *IEEE TRANSACTIONS ON AUDIO AND ELECTROACOUSTICS*, vol. AU-15, no. 2, pp.74-75, 1967.

Cooley, J.W., Lewis, A.W. and Welch, P.D., "Historical Notes on the Fast Fourier Transform," *IEEE TRANSACTIONS ON AUDIO AND ELECTROACOUSTICS*, vol. AU-15, no. 2, pp.76-79, 1967.

Jenkins,G.M. and Watts, D.G., *SPECTRAL ANALYSIS and its applications*, Holden-Day, San Francisco, 1968.

Priestley, M.B., *SPECTRAL ANALYSIS AND TIME SERIES*, Academic Press, New York, 1981.

Welch, P.D., "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms," *IEEE TRANSACTIONS ON AUDIO AND ELECTROACOUSTICS*, vol. AU-15, no. 2, pp.70-73, 1967.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Naval/Mechanical Engineering Curriculum Office, Code 34 1
   Naval Postgraduate School
   Monterey, CA 93943-5101

4. Professor Fotis A. Papoulias, Code ME/PA.. 4
   Naval Postgraduate School
   Monterey, CA 93943-5101

5. LT Gary G. Elvik, USN... 6
   218 Tunisia Rd.
   Seaside, CA 93955

66 290NPG TH 2677
6/02 22527-200 NLB

DUDLEY KNOX LIBRARY
3 2768 00402342 4